Heriot-Watt University

PhD Thesis

Optimising Strategies for Learning Visually Grounded Word Meanings Through Interaction

Author:
Yanchao Yu

Supervisors:
Prof. Oliver Lemon
Dr. Arash Eshghi

A final thesis submitted in fulfilment of the requirements for the degree of PhD

School of Mathematical and Computer Sciences

August 2018

Abstract

Language Grounding is a fundamental problem in AI, regarding how symbols in Natural Language (e.g. words and phrases) refer to aspects of the physical environment (e.g. objects and attributes). In this thesis, our ultimate goal is to address an interactive language grounding problem, i.e. learning perceptual groundings (specifically vision) through Natural Language (NL) interaction with humans. Although some previous work has shown significant progress on language/symbol grounding on different tasks, there are still some limitations and unsolved problems: (a) only learning groundings holistically without understanding individual parts of the linguistic and non-linguistic context, (b) requiring training data of high quantity and quality, but without the possibility of on-line error correction, and (c) not being able to continuously and incrementally learn from the external environment. Most these limitations are likely to be alleviated if systems can learn symbol groundings, as and when needed, from natural, everyday conversations with humans.

For working on all of the above limitations at once, this thesis proposes a modular Interactive Multi-modal Framework, which is *compositional*, *optimised*, trainable *incrementally* with *small amounts of data*, and able to handle *natural, spontaneous dialogue*. Specifically, we collect real human-human conversations (BURCHAK corpus) for investigating how humans behave in an interactive learning task, which contains a wide range of dialogue capabilities, strategies, and linguistic phenomena encountered in natural, spontaneous dialogue. This thesis then explores how different capabilities and strategies (from the real data) affect the overall learning/grounding efficiency, i.e. higher recognition accuracy with less human effort in the dialogue. We found that an agent, that is able to: 1) take initiative, 2) consider both uncertainty from visual classification and context-dependencies from dialogue, and 3) demand further information if necessary, performs better. Finally, following the above results, we train an optimised multi-modal dialogue agent using Reinforcement Learning for addressing interactive language grounding against the real data. The agent learns: (1) to perform a form of active learning, i.e. only ask further information if necessary, and (2) to process natural, daily conversations with humans. Here, we incorporate our framework with an incremental semantic formalism (the DS-TTR framework) that dynamically presents compositional representations for both linguistic and non-linguistic (visual) context, and is able to process natural, spontaneous conversations (specifically incremental phenomena, such as "self-repair").

These advances bring us closer to addressing the interactive grounding problem, and bringing robots from the laboratory into the real world, where they will need to speak in the same language as human beings.

Acknowledgements

Firstly, I want to express my gratitude to my supervisor, Oliver Lemon, for introducing me to the fields of Dialogue System and Human-Robot Interaction, for his guidance and assistance throughout my research life in the past five years (including three-year PhD) and also for providing me invaluable advice. I would also like to thank my second supervisor, Arash Esghi, for always being there to offer critical point of view, new insights and new ideas, for his constant support in the further design, development, and debugging of the DS-TTR model, as well as for helpful criticism and kindness that saves me from my occasional puzzle.

Next, I am very grateful to my Examiner David Schlangen and Frank Broz for their insightful comments at my viva and pointing me to possible future directions, providing me their extensive experience and advice very generously. They have reminded me how much work is still to be done in this domain, which may eventually bring the robot from the laboratory to a real human daily life.

During my three-year PhD, I've had a chance to work with and become friends with wInteraction Lab, at Heriot-Watt University. I've learned a lot from the interaction with them. Many thanks go to my lab colleagues (in alphabetical order): Shubham Agarwal, Amanda Cercas-Curry, Ondrej Dusek, Ioannis Efstathiou, Klaus-Peter Engelbrecht, Dimitra Gkatzia, Helen Hastie, Srini Janarthanam, Simon Keizer, Yannis Konstas, Xingkun Liu, Katrin Lohan, Jekaterina Novikova, Yiannis Papaioannou, Jose Part, Verena Rieser and Igor Shalyminov. I would like to thank Gregory Mills for his support in the dialogue collection experiment with his DiET chat tool. I would also like to thank Andrea Vanzo for our recent discussion of the possible employment and further extension of the interactive learning framework I proposed in this thesis.

Here, I would like to briefly thank the people outside the academic area who have given their encouragement in helping me complete this PhD. I have been fortunate to have thank my best friend, Stuart Gordon, for always being around to provide support and encouragement inside and outside the academic life. I would also like to express my gratitude to my wife Menglu and my daughter Luyi, who made amazing support and encouragement to help me survive this. They have provided me a warm and loving environment at home. Last, but not least, this thesis is dedicated to my parents and in-laws for always being supportive, optimistic and encouraging and for always being on my side. I would not be where I am today without their supports and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We begin with an example from a practical robotics application: imagining one day, we brought a robot back home from the laboratory. In the morning, we ask it to help us bring something, like the conversation below:

User: *Hey, can you bring me my morning mug please?*
Robot: *Sorry, I don't know it. What is a morning mug?*
User: *The mug on the table, black with a yellow smiling face on it.*
Robot: *You mean this one?*
User: *Yes, bring it to me.*
Robot: *Okay, here you are.*
User: *Thanks.*

In the beginning of this conversation, the robot did not complete the command/task, because it cannot understand what is a "morning mug", until we explained what the mug looks like. In this case, the "morning mug" is meaningless to the robot until it is able to map this to a real object in the room. Such an issue with the robot is commonly called 'Symbol Grounding'. Symbol Grounding [1] is a fundamental problem in AI and cognitive science about how symbols in a language can refer to objects and properties in the external world (Harnad, 1999). In this thesis, our research focuses on addressing the problem of *interactive* language grounding, i.e. learning how symbols of a language are grounded in perception, specifically vision, through Natural Language (NL) interaction with human tutors.

In recent years there has been a surge of interest and significant progress made on a variety of related tasks, for instance, researchers attempt to either generate NL descriptions for images/videos, or in the opposite way, i.e. identify/retrieve them following certain NL descriptions (Karpathy and Li, 2015, Silberer and Lapata, 2014, Socher et al., 2014). Most of these works address the problem by performing a form of holistic/implicit grounding

[1] The term 'grounding' is also used in dialogue research to mean when 2 or more speakers agree on the content of a conversation so far, which is called communicative grounding (Clark and Brennan, 1991).

approaches, which usually project holistic or partially compositional representations from different modalities (vision and language) into a common space, and apply variety of neural modelling methods to discover their association to retrieve or generate one from the other. Although these works have achieved good performance, they did not address the classic symbol grounding problem as described by Harnad (1999). In that paper Harnad (1999) emphasises that the symbol system should be semantically interpreted following *"an explicitly represented symbolic rule, which is part of a formal system and decomposable"*. The explicit representation, consisting of elements, can be recombined in systematic ways. In contrast with prior work, in this thesis, we aim to address compositional grounding, i.e. learning the alignment between parts of user commands/utterances and different elements of the visual scene (e.g. colour and shape properties) from the external world, similar to Kennington and Schlangen (2015), Matuszek et al. (2014).

In addition, as part of the grounding problem, an increasing amount of recent work, such as Karpathy and Fei-Fei (2014), Kiros et al. (2014), Ngiam et al. (2011), Socher et al. (2014), has shown good progress on resolving visual classification issues using Deep Neural Networks, which normally require large amounts of data for training. However, although training with such large data can provide higher accuracy than other machine learning approaches, there are two key limitations: 1) highly expensive user annotations, i.e. a collection of high-quality visual examples always requires a large number of annotators spending significant effort on segmenting images and manfully adding corresponding labels/descriptions; 2) high quality of visual/textual representations, especially with complex visual scenes, i.e. more complicated visual scenes atypically-shaped objects and even new concepts/symbols are still not accurately represented or annotated. Hence, instead of using large amounts of annotated visual data, we focus in this thesis more on learning visual categories incrementally, starting with little or no knowledge. In this chapter, we define an interactive visual-attribute learning task, in which a learning agent needs to learn visual knowledge from scratch, by learning visual classifiers that can be updated in dialogue with a human tutor over time. This is different from existing systems such as (Bruni et al., 2014, Silberer and Lapata, 2014) which have robust, prior visual knowledge (i.e. pre-trained visual classification models) that aim to map their existing knowledge to Natural Language.

Last but not least, there are some previous approaches that, instead of learning through NL interaction, learn groundings from images/videos pre-annotated with NL descriptions or definite reference expressions or following a series of pre-defined rules, for instance, Kennington and Schlangen (2015), Socher et al. (2014). However, we argue that, in more complicated situations with massive variation and uncertainty, NL interaction can provide further help to the system in the learning task, because of its capability of online error correction, i.e. assisting the learner/system to correct any mistakes on the pre-defined labels or its own

predictions of particular visual scenes through real-time conversations. Additionally, we acknowledge that one of the biggest challenges in Human-Robot Interaction is that an individual person is used to describe the visual scene (e.g. objects, attributes and even events) from a personal perspective. It means that, instead of learning general annotations, the system is more expected to learn them under personal descriptions, for instance, the "morning mug" described above. Communicating with users may lead to further information (e.g. confirmation, declaration and explanation) that prompts it to process (understand and generate) individual descriptions following the user preferences. On the other hand, different with synthetic or agent-agent dialogues that contain *well-structured* utterances, human daily, spontaneous conversations always contain a large/broad variety of user expressions and also a wider range of dialogue phenomena (such as self -repair, -repetition, hesitation, continuation, etc.). Those dialogue expressions and phenomena might negatively impact on the quality of utterance understanding, the next system moves/responses, as well as lead to task failure. In this thesis, we therefore explore an appropriate model for coping with such dialogue phenomena within conversations. We are also concerned with an optimised dialogue strategy in support of effectively learning visual groundings by processing natural, incremental conversations with human users.

1.1 Research Questions & Contributions

In order to fulfil the motivation described above, there are several research questions explored in this thesis:

- **Research Question 1**: What visual classification models are better suited to the problem of learning incrementally from small amounts of data? *(Chapter 4)*
- **Research Question 2**: What are the important characteristics of spontaneous human dialogue in interactive concept learning/teaching? *(Chapter 5)*
- **Research Question 3**: Given the learning task, interactive systems that learn continuously, and over the long run from humans are expected to do so incrementally, quickly, and with minimal effort/cost to human tutors. So instead of simply copying human behaviours, how can a system/robot effectively learn novel knowledge from humans but with less human effort in such a learning process? *(Chapter 7 and 8)*
- **Research Question 4**: Does the capability of processing incremental phenomena (e.g. "self-repair") improve the overall grounding/learning performance of the agent given an interactive grounding task? *(Chapter 9)*

For answering those questions, in this thesis, we design and implement an Interactive Multimodal Framework, which is *compositional*, *optimised*, trainable *incrementally* with *small amounts of data*, and able to handle *natural, spontaneous dialogue*.

More specifically, we explore a more appropriate visual classification model (Logistic Regression SVM with Stochastic Gradient Descent (SGD-SVM)) for learning low-level perceptual features (colours and shapes) with a small amount of data incrementally over time. We attempted to consider the visual classification task as a data-driven task from diverse ways, for instance, multi-label classification, offline and online/incremental learning. Since these classification approaches are associated with different conceptual and practical foundations and have been both applied for Object Recognition, we directly compare these models given a visual-attribute learning task. Through a series of experiments (see Yu et al. (2015a,b)), the SGD-SVM model is found to be more desirable for the interactive learning task than the other methods: it achieves higher accuracy of attribute prediction, and compositionally learns different attributes from each single object faster than the others.

In addition, we present a new Human-Human dialogue corpus for interactive learning of visually grounded word meanings through ostensive definition by a tutor to a learner (Yu et al., 2017b). As mentioned earlier, a robot that is brought from the laboratory to the external world is expected to behave like a real human, i.e. talking about the visual environment just like how a human does. Hence, different with a Wizard-of-Oz (Woz) technique (Fraser and Gilbert, 1991) that is commonly used to investigate how humans interact with a robot/machine, a human-human experiment brings more benefits on the investigation of realistic human behaviours in both roles of participants, i.e. given an interactive attribute learning task, it allows us to investigate, not only about how people would teach novel knowledge but also about how humans as the learner can acquire useful information, through natural, daily conversations. Riek (2012) and Li and Dey (2013) emphasised that one of the difficulties in the Woz method is that some features in the human behaviour, e.g. complex decision-making and unexpected errors or mistakes, cannot be easily simulated by the Wizard (machine).

Given an interactive visual attribute learning task, participants are assigned to different roles (e.g. tutor and learner) who teach/learn to describe visual objects in a made-up language (e.g. "burchak" for square, "sako" for red) by communicating with each other. Although the corpus contains only 177 dialogue examples, it surprisingly reflects a lot of variation on dialogue expressions and strategies given such a relatively simple task. The most challenging aspect of this corpus is that it contains a wide range of linguistic phenomena encountered in natural, spontaneous dialogue [2], for instance, self -repair and -repetition, filler, overlap and

[2]To our knowledge, this corpus is the first realistic human-human dialogue collection on the interactive visual-attribute learning domain, which contains a wide range of natural spontaneous dialogue phenomena.

etc. We believe that both the dialogue strategies and incremental phenomena investigated in this corpus are likely to impact on the learning/grounding performance in this research.

Following the analysis of realistic human-human conversations on the interactive grounding problem, we explore how different dialogue capabilities and strategies (from both sides of the tutor and the learn) influence the learning/grounding efficiency (Yu et al., 2016b,c,d). Given the interactive learning task, we expect a better dialogue/learning strategy that supports the agent to achieve and maintain a relative balance between the learning performance (classification accuracy) and a tutoring/dialogue cost: learning unknown visual knowledge more accurately, but with less dialogue effort by the human tutor. In terms of the learner's behaviours (the agent in this research performs as a learner), we mainly take into account three essential dialogue properties: (1) who takes *initiative* in the dialogues; (2) the agent's ability to utilise their level of *uncertainty* on visual attributes; and (3) context-dependency, i.e. their ability to process elliptical as well as incrementally constructed dialogue turns. On the other hand, we are concerned more with potential learner's strategies responding to the tutor's behaviours: knowledge-demanding, i.e. a basic ability to request further feedback or information when the tutor did not provide enough knowledge in the previous conversation. Our experiments show that differences along these dimensions have significant impact both on the accuracy of the grounded word meanings that are learned, and the processing effort required by the tutors. In order to achieve a better trade-off between the learning performance and dialogue cost, we show that the agent should take the initiative in all dialogues, and also take into account classification uncertainty and context-dependency. Meanwhile, given a "lazy" tutor who does not always provide all the information about the visual scene, taking into account a strategy of knowledge-demanding may lead to more opportunities to improve the learning performance.

Finally, according to investigations and explorations from the previous work, we ultimately manage to build and train an optimised learning dialogue strategy using Reinforcement Learning with a multi-objective Markov Decision Process (MDP) (Yu et al., 2017a). It successfully achieves a learning agent as we motivated above: the agent learns to 1) perform a form of "*active learning*" (i.e. where the system only asks further feedback (information) from a human tutor when it is uncertain about the correctness of its predictions of visual attributes), and 2) takes the initiative within conversations and processes natural, human-like conversations. For processing the incremental, spontaneous dialogue investigated in the BURCHAK corpus (Yu et al., 2017b), instead of applying a simple Dialogue Act tagging model (SimpleSLU), we deploy an incremental, word-by-word parser – DyLan (Eshghi et al., 2011) – that is implemented based on the DS-TTR (Dynamic Syntax and Type Theory with records) formalism (Eshghi et al., 2012, Hough, 2011, Purver et al., 2014), which not only dynamically presents compositional representations for both linguistic and non-linguistic

(specifically visual) context, but also has shown good performance on processing incremental phenomena in natural, spontaneous conversations, such as "self-repair". We introduce a mechanism of dialogue act inference that automatically predicts the most appropriate dialogue act based on certain completed semantic sub-trees following a set of pre-learned rules in the DS-TTR framework (see more in Chapter 9 and Appendix A). Through experiments (see Chapter 9), we show that the new DS-TTR agent is able to keep a comparably good performance on the learning/grounding task, especially while interacting with an incremental simulated tutor (which, trained based on realistic data from the BURCHAK corpus, can randomly generate incremental dialogue phenomena (see Chapter 6)). We note that that this is the first time that the DyLan parsing model has been implemented for coping with realistic human-human dialogues.

1.2 Overview of Thesis Structure

As described above, this research contributes to a robust and modular framework of interactive language grounding in support of building interactive multi-modal teachable systems. The rest of this thesis is structured as below:

Chapter 2 mainly reviews the previous work for addressing the visual language grounding task. The review can be generally identified into two key parts. In the first place, we introduce the symbol grounding problem in AI and cognitive science, and also a surge of previous work that has approached the grounding task from three dimensions: 1) performing *holistic* or *compositional* grounding. 2) through *interaction* or *offline*, and 3) using *hand-crafted* or *optimised* dialogue management. On the other hand, we also provide an overview of dialogue processing and relevant methods, where we mainly focus on incremental conversational systems given the learning task mentioned above.

Chapter 3 introduces a novel interactive multi-modal framework in support for building a teachable robot/interface that learns the alignment between NL symbols and the visual scene in the physical environment through dialogues. We also briefly introduce a list of key components and relevant approaches, for instance, classification approaches for the vision module, NL understanding and dialogue management for the dialogue module. In addition, given the interactive learning domain, we also introduce a simulated learning environment as part of the framework for training and evaluating the interactive system/robot. In this thesis, both the simulated environment and the framework are applied to explore the best conversational learning policy for addressing the interactive learning problem.

Chapter 4 compares the state-of-the-art approaches to visual attribute-based classification through NL interaction with humans: 1) offline binary approaches (e.g. linear SVM models);

2) multi-label approaches, as well as 3) an online classification approach (also known as incremental learning methods). The performance of each learning model in an interactive learning process is evaluated through experiments.

Chapter 5 motivates and describes a new freely available human-human dialogue data set on an interactive visual-attribute task, where participants (assigned to a tutor or a learner) learns visually grounded word meanings in a made-up language by communicating with each other. The text-based interactions (via a novel, character-by-character variant of *the DiET chat tool* Healey et al. (2003), Mills and Healey (2017)) closely resemble face-to-face conversation and thus do not only contain a lot of variations on dialogue capabilities and strategies, but also contain a wide range of the linguistic phenomena encountered in natural, spontaneous dialogue, including self- and other-correction, mid-sentence continuations, interruptions, overlaps, fillers, and hedges. The realistic dialogues in the corpus are applied to train a natural dialogue system.

Chapter 6 introduces a new freely available framework for building accurate user simulations, which also simulate natural incremental phenomena such as self-repairs. Differently to other existing user simulations, the challenge for this framework is that it aims at not only resembling user strategies and capabilities in realistic conversations, but also at simulating natural incremental dialogue phenomena, e.g. self-repair and repetition, and pauses, as well as fillers. The proposed simulation method is evaluated on two different dialogue corpora, i.e. the BURCHAK corpus from Chapter 5 and the Facebook bAbi corpus (Weston et al., 2015) (a synthetic dialogue dataset).

Chapter 7 designs and compares different dialogue capabilities and policies for interactively teachable systems collected from the BURCHAK corpus. It does not only consider Tutor behaviours (i.e. "Good"/"Lazy" tutor policy), but also takes the learner's dialogue strategy into account. The learner dialogue strategy refers to a variety of dialogue capabilities on the human learner side, including initiative, uncertainty and context-dependency. The overall performance (i.e. a combined measure of recognition performance and cost) of each condition is explored to find the most appropriate dialogue strategy for interactive learning tasks.

Chapter 8 presents an optimised multi-modal dialogue agent for interactive learning of visually grounded word meanings from a human tutor, trained on the real human-human tutoring data. Within a life-long interactive learning period, the agent, trained using Reinforcement Learning (RL), must be able to handle natural conversations with human users, and achieve good learning performance (i.e. accuracy) while minimising human effort in the learning process. Our experiment has demonstrated that, the agent can 1) process coherent conversations with the simulated user to achieve the goal of the task (i.e. learning visual attributes of different objects, e.g. colour and shape); and 2) finds a better trade-off between

classifier accuracy and tutoring costs than hand-crafted rule-based systems, including ones with dynamic policies.

Chapter 9 proposed a new optimised learning agent, as an extension of the previous system in Chapter 8, by incorporating with an incremental word-by-word semantic parser (DyLan) instead of the hand-crafted dialogue act tagging model. In this chapter, we extend the existing DyLan parser to infer appropriate dialogue acts by mapping completed semantic sub-trees to a specific user intent (or dialogue act) by executing particular computational actions. Through the experiment, the newly proposed system shows comparably good performance on processing natural, incremental conversations with the human tutor through the learning period and also keeps achieving a better balance between the accuracy and the cumulative tutoring cost.

1.3 Publications

The following are the publications presented from this work:

1. Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2015 *Interactive Learning through Dialogue for Multimodal Language Grounding.* In Proceedings of the 19th SEMDIAL, Gothenburg. (*associated with Chapters 3 and 4*)

2. Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2015 *Comparing attribute classifiers for interactive language grounding*, In Proceedings of the 4th VL Workshop, Lisbon. (*associated with Chapters 4 and 3*)

3. Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2016. *Comparing dialogue strategies for learning grounded language from human tutors.* In Proceedings of the 20th SEMDIAL, New Jersey. (*associated with Chapters 3 and 7*)

4. Yanchao Yu, Arash Eshghi, and Oliver Lemon, 2016. *An Incremental Dialogue System for Learning Visually Grounded Language (demonstration system)*, In Proceedings of the 20th SEMDIAL, New Jersey. (*associated with Chapters 3 and 7*)

5. Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2016. *Interactive Learning of Visually Grounded Word Meanings from a Human Tutor.* In Proceedings of the 5th VL Workshop, Berlin. (*associated with Chapters 3 and 7*)

6. Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2016. *Training an adaptive dialogue policy for interactive learning of visually grounded word meanings.* In Proceedings of the 17th SIGDial, Los Angeles. (*associated with Chapters 3 and 7*)

7. Yanchao Yu, Oliver Lemon, and Arash Eshghi. 2016. *Incremental Generation of Visually Grounded Language in Situated Dialogue (demonstration system).* In Proceedings of the 9th INLG 2016, Edinburgh. (*associated with Chapters 3 and 7*)

8. Yanchao Yu, Arash Eshghi, Gregory Mills and Oliver Lemon. 2017. *The BURCHAK corpus: a Challenge Data Set for Interactive Learning of Visually Grounded Word Meanings.* In Proceedings of the 6th VL workshop, Valencia. (*associated with Chapters 5 and 6*)

9. Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2017. *Learning how to learn: an adaptive dialogue agent for incrementally learning visually grounded word meanings.* In Proceedings of the 1st RoboNLP, Vancouver. **(Best Paper Award)** (*associated with Chapter 8*)

10. Yanchao Yu, Arash Eshghi, and Oliver Lemon. 2017. VOILA*: An Optimised Dialogue System for Interactively Learning Visually-Grounded Word Meanings (Demonstration System).* In Proceedings of the 8th SIGDial, Saarbrücken. (*associated with Chapter 8*)

Chapter 2

Related Work

As introduced in the previous chapter, the ultimate goal of this thesis is to address interactive visual grounding problem. In such a target, our work lies exploring and implementing an appropriate approach in a visual-attribute learning task, in which an agent/robot needs to learn how to identify and describe visual objects using their attributes (e.g. 'red', 'black' for colour and 'square', 'triangle' for shape), from scratch, incrementally through daily conversations with real humans. It is required to drive an effective conversation to gain useful information from human tutors, by asking suitable questions, for example, "can you tell me what colour this object is?" or "I think it is a red square, isn't it?"

In order to achieve such a goal, the agent needs to be able to understand and generate *compositional* language, and should be *optimised* to learn *visual-attributes* through natural language interaction with human partners *incrementally*, over time. Different to experimental systems/interfaces, a robot brought into the real world also needs to be capable of processing *natural, spontaneous conversations* with human partners, which involves more complicated dialogue phenomena with contains many variations and uncertainties. These properties will play an important role on the exploration of appropriate approaches in support of building such robust learning agents for the grounding problem in the rest of this research.

Before exploring the potential architectures and methods for the problem, in this chapter, we step back to review some of the history of the symbol grounding problem. After that, we will have a look at previous work to investigate how it addresses such grounding problems from different aspects: 1) which properties/features of this problem they are addressing, and 2) what architectures and approaches have been deployed for this problem. On the other hand, as our work mainly focuses on the impact of NL conversation with humans on grounding success/performance, we will also briefly review spoken dialogue systems and relevant methods for processing natural, everyday conversations with real humans.

Specifically, this chapter will contribute across the following sections:

1. Section 2.1 introduces the symbol grounding problem.
2. Section 2.2 and 2.3 present some previous work, which has shown significant progress on the perceptual language grounding for robotics, as well as their related architectures and approaches for the grounding problem. Here, we discuss these advances on three main dimensions: 1) learning a *holistic* or *compositional* alignment/grounding between the visual scene and language, 2) learning from pre-annotated visual examples (*offline*) or through *interaction* with other agents or humans, as well as 3) interactively learning the grounded word meaning via *hand-crafted* or *optimised* strategies.
3. Section 2.4 takes a look at the background of dialogue systems (e.g. situated dialogue and incremental dialogue systems). It also discusses the suitability of the classical dialogue system framework for incrementality in natural, everyday conversations. At the end of this section, we briefly introduce an incremental formalism for processing dialogue (called Dynamic Syntax (DS)), which will be employed and extended as an essential part of this research work.

Finally, section 2.5 will summarise our investigation through a set of existing researches, not only on the grounding problem itself, but also on the field of dialogue processing, where we will discuss distinctions between our work in this thesis and previous work.

2.1 The Symbol Grounding Problem

> "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols."
> – Harnad (1999)

Given the description from Harnad (1999) above, the symbol grounding problem can be understood as the problem of assigning meaning to symbols by associating them with the external world.

Here we briefly discuss the possible ways in which a robot could ground symbols by building such connections. Before considering promising solutions for the symbol grounding problem

for robots, we would like to briefly explore how a human might cope with such a grounding task. Harnad (1999) indicated three human behavioural capacities as core elements for addressing the symbol grounding problem: 1) *iconization* – that is an ability to transform "the projection of distal objects on humans' sensory surfaces (Shepard and Cooper, 1986)" to iconic representations (or perceptual representations to the computer); 2) *discrimination* – that, as a relative judgement based on human capacity, distinguishes different iconic inputs (from either vision or language) and also illustrates how different they are; as well as 3) *identification* – that is viewed as an absolute capacity to judge whether a given input can be assigned with a unique name, a member of a specific category (where the symbols are eventually grounded). This scheme will be similar to how a robot needs to perform for the grounding problem. Harnad (1999) also claimed that the capability of discrimination and categorisation can eventually contribute to an ability of describing/responding to descriptions through symbolic representations for both human beings and robots.

2.2 Visual Language Grounding in AI & Computational Linguistics

Following the literature on symbol grounding, visual language grounding is one of its subfields that focuses on producing the alignment between the meaning of symbols in Natural Language (e.g. words, phrases and sentences) and aspects of the visual environment (for instance, objects, and events). Learning such alignment/grounding is viewed as an crucial challenge in the fields of Artificial Intelligence (AI) and Computational Linguistics.

Regarding the theory of symbol grounding, Harnad (1999) emphasised that, although many phenomena in the real world can be semantically interpreted, some of them are not symbolic. The boundary between symbolic or not is formulated based on whether a phenomenon is interpreted as following an *explicit* or *implicit* rule [1]. An example from Harnad (1999) is: A thermostat may be interpreted as following the rule: Turn on the furnace if the temperature goes below 70 degrees and turn it off if it goes above 70 degrees, yet nowhere in the thermostat is that rule explicitly represented. According to Harnad (1999). an explicitly represented symbolic rule is decomposable, whose application and manipulation should be purely syntactic, shape-dependent, i.e. "being symbolic must be a systematic property." Based on this theory, the explicit (compositional) "rule" will play an essential role on addressing the visual grounding task in our research.

[1] The main distinction between explicit and implicit rules is that: "It is not the same thing to "follow" a rule (explicitly) and merely to behave "in accordance with" a rule (implicitly)."(Wittgenstein, 1953)

Regarding the grounding problem with robots, based on the theory of symbol grounding from Steels et al. (2005) and Belpaeme (2001), instead of learning the meaning of symbols by the robot alone (either incrementally through an increasing number of examples or not), communication or feedback from others (e.g. humans or robots) might bring positive impacts on the grounding problem, especially within more complicated situations with a variety of interrelations and uncertainty, where knowledge exchange with other agents or humans is likely to provide more clear explanations on what happens either in the entire visual scene or between individual segments. Hence, in the case of addressing the grounding task, such communication is also viewed as an important feature of the learning agent.

Based on discussions above, in this section, we will mainly focus on addressing the grounding problem from three core dimensions: 1) whether they learn the alignment between vision and language within a ***holistic*** (implicit) or ***compositional*** (explicit) process ; 2) whether their work learns such vision-language alignment from visual examples with pre-annotated NL descriptions (***offline***) or through ***real-time conversations*** with other agents or humans; and 3) whether these models/approaches cope with their learning and interaction behaviours via ***hand-crafted*** rules or through a trained, ***optimised*** dialogue policy. we review some previous work which has shown significant progress on addressing the grounding problem. We will investigate and discuss what architectures and approaches they applied to this problem, and also judge how many properties they have taken into account in their models. More details are presented in the following sections.

2.2.1 Holistic versus Compositional Grounding

In the field of visual grounding, we can investigate an explicit process, called 'compositional grounding' (Greco and Carrea, 2012), that learns to align elements of Natural Language (e.g. words or phrases) with aspects of the visual scene (e.g. specific person, visual object, or attribute). The difference between holistic and compositional grounding is that compositional grounding uses explicit representations from both modalities (e.g. vision and language) but implicit grounding does not. The explicit representations, consisting of elements, can be recombined in systematic ways, but the implicit representations cannot (Rougier, 2009).

Recently, there has been a large volume of work that addresses the grounding problem in "holistic" (implicit) ways: in this category of work is the large literature on image and video captioning systems that learn to associate an image or video with NL descriptions (Karpathy and Li, 2015, Silberer and Lapata, 2014, Socher et al., 2014). This line of work uses various forms of neural modelling to discover the association between information from

multiple modalities. This often works by projecting holistic representations from the different modalities (e.g. vision and language) into the same space in order to retrieve or generate one from the other. Here, we briefly describe some of these projects, as below:

Silberer and Lapata (2014) propose a multi-modal approach (using an Auto-encoder) to learn grounded meaning representations by mapping both words/concepts and visual objects into common embedding spaces (see model in Fig. 2.1). They point out that this model can be trained with a semi-supervised objective (Silberer and Lapata, 2014). Its input modalities consist of vector-based (distributional) representations [2] of words and outputs (confidence scores) of visual classifiers from images. It trains a model with two hidden layers for each input modality separately and then yields a fused meaning representation by joining them together.

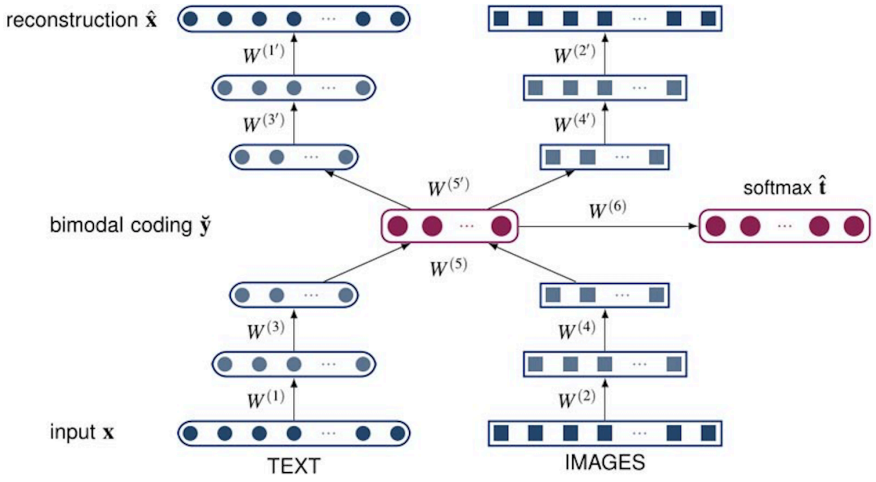


FIGURE 2.1: Illustration of Auto-encoder trained with Text and Images (Silberer and Lapata, 2014)

Socher et al. (2014) propose a multi-modal learning model, called Dependency-Tree Recursive Neural Network (DT-RNN), that projects a sequence of words $w_1, ..., w_m$ in a sentence into a DT structured representation (a compositional semantic representation) to explore parents of each node and correlations between nodes. The main idea of this work is to store images and sentences in the same multi-modal space, so that similar images or descriptions can be queried/retrieved with the model by exploring the nearest neighbours (see Fig. 2.2). The model is trained with N images with corresponding feature vectors (a distributional representation) of each visual scene (image) and DT-structured semantic representations of formal descriptions.

Karpathy and Li (2015) propose an alignment model for inferring the relations between visual and textual data using a multimodal Recurrent Neural Network (see diagram in Fig. 2.3). Instead of the conventional dependency-tree parser, sentences can be represented

[2] The dimensions of the representations correspond to textual and visual attributes

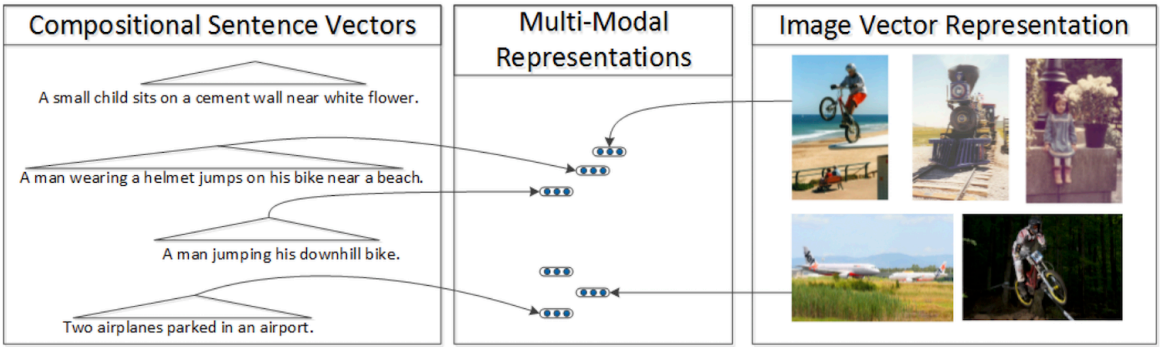


FIGURE 2.2: Illustration of DT-RNN model based on Dependency Tree (Socher et al., 2014)

with a bidirectional Recurrent Neural Network (BRNN) that considers both single words as well as their sequence and relevant context information within the sentences. The BRNN model involves two independent processes: one of them encodes a sequence of N words into a $1-of-k$ representation and then converted to an h-dimensional vector v, and another one embeds object regions in an image using CNN models. This model is trained to assign both word vectors and object regions to the same location π_t to log a list of align scores for image-sentences pairs. In contrast to Socher et al. (2014)'s work that learns to retrieve either similar images or descriptions given the other sources, Karpathy and Li (2015)'s model generates a final description for novel images also based on these correspondences.

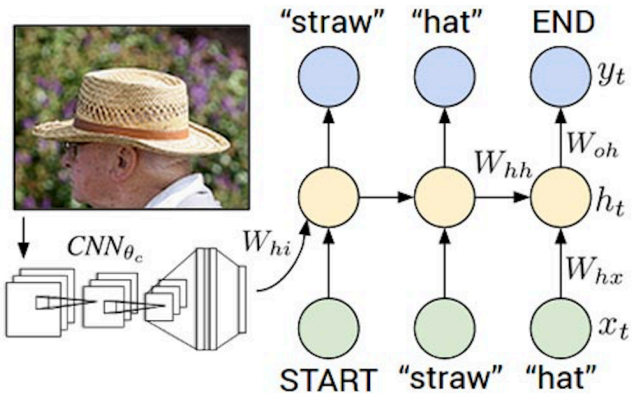


FIGURE 2.3: Diagram of the Multi-modal Recurrent Neural Network (Karpathy and Li, 2015)

Although these approaches have shown good performance on resolving the grounding problem to generate or retrieve NL descriptions, their grounding approaches would not be symbolic based on Harnad (1999)'s account. On the contrary, other models assume a more direct/explicit connection between symbols (either words or predicate symbols of some logical language) and perceptions Kennington and Schlangen (2015), Kollar et al. (2013), Matuszek et al. (2014, 2012), Tellex et al. (2014, 2013), Yu et al. (2016c). In this line of work, representations are both compositional (symbolic) and transparent, with their constituent

atomic parts grounded individually in perceptual classifiers. Our work in this thesis is in the spirit of the latter.

Matuszek et al. (2014, 2012) propose a system that automatically learns the meaning of different visual attributes using a language-perception joint model. This work is more concerned with finding the correct object based on user's descriptions/commands, than learning novel objects. This model aligns the logical constants within a logical form z and a set of classifiers in the set C. The logical form z is parsed into a λ-calculus expression using the Combinatory Categorical Grammar (CCG) semantic parser (Steedman, 1991). For instance, a description like "this red block is in the shape of a half-pipe" can be represented into $\lambda x \cdot shape(x, arch) \wedge colour(x, red)$ (see Fig. 2.4).

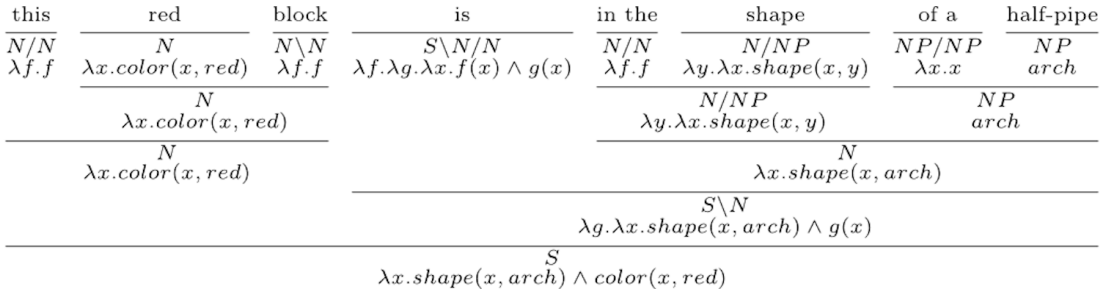


FIGURE 2.4: Example of a λ-calculus logical form in CCG (Matuszek et al., 2012)

This joint model is trained to pair the logical constant *red* or *arch* into a specific classifier $c \in C$ using a combination of a feature vector (produced by bag-of-words) from a set of words/logical forms W, and the output of the visual classifiers for each object. To find the correct object, the logical form can be executed by scanning all possible objects with estimated likelihood γ_o until finding one for which all classifiers return true. This work is closest to our work in this thesis, where, instead of using CCG semantic parser (Steedman, 1991), we deploy the DyLan model (an incremental word-by-word parser/generator, incorporated with Type Theory with Records (TTR)) to produce the semantic analysis of both visual and linguistic context (see more details in Section 2.4.3).

Tellex et al. (2014, 2013) infer the alignment of NL constituents with a given set of NL commands and corresponding video examples using a proposed Probabilistic Graphical model G^3 (see Fig. 2.5) which maps parts of NL commands to objects, places, paths and even events in the external world. They implement a system using the G^3 model that allows for asking questions by detecting grounded variables ($\gamma_i \in \Gamma$) with uncertainty values (Tellex et al., 2014, 2013). The most uncertain variable γ_i can be found using a distribution with the highest entropy. The system also supports a determination of when to ask questions or take actions using a certain threshold for the entropy estimation, for instance, the system will ask a question while the highest entropy is lower than a certain threshold.

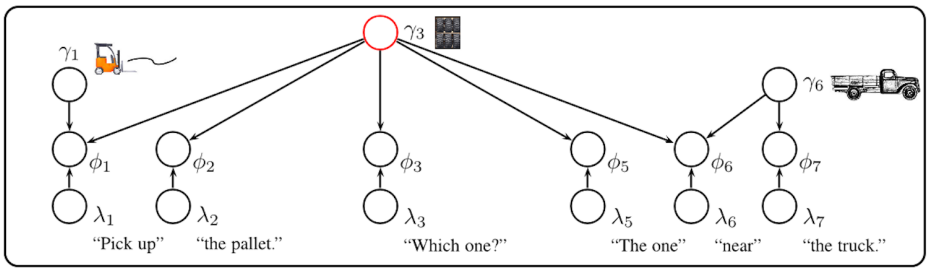


FIGURE 2.5: The Grounding Graphs for a three-turn dialogue using G^3 model (Tellex et al., 2014, 2013)

Kollar et al. (2013) propose a LSP (Logical Semantics with Perception) model that aims at jointly learning semantic parsing NL sentences to both logical forms and perceptual classifiers. Similar to Matuszek et al. (2014)'s work, the LSP model also deploys the CCG parser to parse statements into logical forms. It contains three main components – perception learning, and parsing – as well as evaluation, as illustrated in Fig. 2.6). This model may eventually generate a denotation and a grounding via a combination of a logical form and logical knowledge base specified by a set of perception classifiers. This model supports both full and weakly supervised learning .

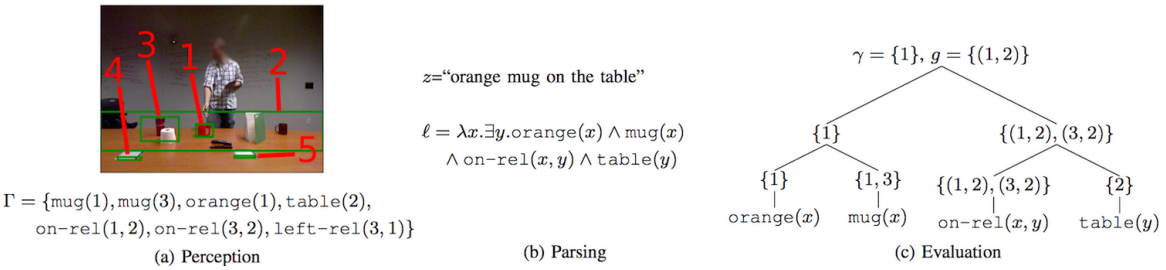


FIGURE 2.6: Illustration of LSP's three components (Kollar et al., 2013)

Both G^3 (Tellex et al., 2014, 2013) and LSP (Kollar et al., 2013) models address grounding problems between NL constituents and physical entities/relations as a set of latent correspondence variables, which is similar to what we intend in the project. Moreover, similar to these approaches, which parse NL sentences into logical forms using CCG (Steedman, 1991), we implement a multi-modal framework with a state-of-the-art incremental parser (DS proposed by Purver et al. (2011)) that parses NL sentences into a form of Type Theory with Records (TTR) (Dobnik et al., 2012a) word-by-word. We will explain how the DS-TTR model is applied to represent the non-linguistic context in Chapter 3

Apart from these approaches described above, (Kennington and Schlangen, 2015) learn the mappings between visual features and the target words directly. In contrast to more conventional grounding solutions, Kennington and Schlangen (2015) define the problem into individual tasks, which learn a mapping between individual words and low-level visual features (e.g. colour-value), and compose the evidence into classifier predictions for a full

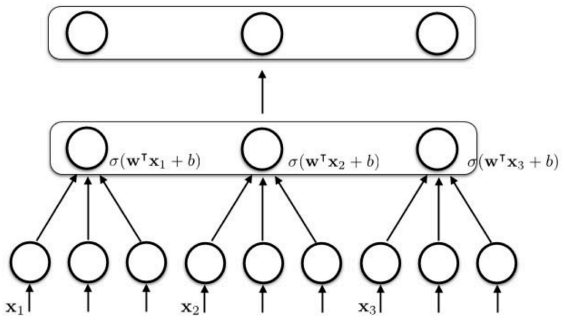


FIGURE 2.7: Representation of the Multi-class Network with Normalisation Layer (Kennington and Schlangen, 2015)

expression. Through this approach, each word w can be referred to a binary logistic regression SVM classifier with a weight vector: $p_w(x) = \sigma(W^T x + b)$, where σ is the logistic function. Afterwards, to predict a pair of word and visual object, a multi-class network is produced containing multiple applications of the individual logistic network for each word (see example in Fig. 2.7). The output of these logistic functions are normalised across all candidate objects. According to Kennington and Schlangen (2015), this model can learn not only words for individual objects but also relations between each other using the pair of a *landmark* and a *target*.

Our work in this thesis is similar to Kennington and Schlangen's work (2015). However, our research represents the words/phrases/sentences in a logical form (TTR record types) instead of using words. The logical form is strictly compositional, i.e. the contribution of the meaning of an individual word, or semantic atom, to the entire representation is clear. Meanwhile, we focus more on learning the pair of semantic representations and visual objects/attributes through interaction with real humans than from textual descriptions or manual annotations.

2.2.2 Interactive versus offline Learning

Another dimension along which work on grounding can be compared is whether groundings are learned from images or videos pre-annotated with NL descriptions or definite reference expressions as in Kennington and Schlangen (2015), Socher et al. (2014)), or from live interaction as in e.g. Das et al. (2016), de Vries et al. (2016), Skocaj et al. (2016), Thomason et al. (2016a, 2015), Yu et al. (2015a, 2016c). In this section, we will mainly present relevant work that addresses the grounding problem through NL conversations, as below:

Thomason et al. (2016a, 2015) present a multi-modal language grounding model that learns a mapping from semantic meanings to visual objects and attributes with an interaction task called "I Spy", in which the agent is required to guess what the user was describing through

NL dialogue. The main idea of this research is to align the predicates from NL descriptions to specific objects by rewarding/penalizing the agent's behaviour. Given a game O_T, the agent assigns scores S to each visual object $i \in O_T$ on the table, where the score is formulated: $S(i) = \sum_{p \in H_p} G_p(i)$ where the agent guesses objects in descending order by scores S, by asking whether the chosen object is correct. Once it is correct, the chosen object will be applied as a positive sample to train classifiers for further predicates. On the other hand, the robot/agent is allowed to describe objects based on knowledge from the previous games. The agent can be rewarded with a predicate score $R(p) = |O_T|\, G_p(i^*) - \sum_{j \in O_T \ \{i^*\}} G_p(j)$ while correctly describing the chosen object i^*, or is otherwise penalized.

Das et al. (2016) introduce the task of "Visual Dialogue" (VD), in which the agent needs to handle a Natural Language conversation (i.e. pairs of question-answer) with humans about specific visual content (i.e. an image). Specifically, this task requires the agent to ground the question from humans into that image/visual content or retrieve information to answer the question. For finding the best VD models Das et al. (2016) implemented and compared a number of neural networks with different LSTM-based encoder-decoder combinations. The encoder, such as a late fusion model, hierarchical recurrent model, or memory network, is applied to generate a joint representation (distributional/vector representation) for the VD model inputs, including image I, dialogue history H, and question Q_t. On the other hand, the decoder (e.g. generative LSTM and discriminative softmax model) will find out the best "ground-truth" answer sequence given the certain encoded vector representation by ranking candidates upon their log-likelihood scores (see more details in (Das et al., 2016)).

Similar to Das et al. (2016)'s work, de Vries et al. (2016) also proposed a LSTM-based intelligent agent for grounding Natural Language into a perceptual environment through a novel visual question-answering (QA) game – GuessWhat?! – that assigns two participants, including a questioner and a oracle. Within the game, given an image with K segmented visual objects, the questioner (or the agent) is required to locate the correct object with the oracle (a human player) by asking polar questions, and the oracle is only allowed to answer "yes" or "no". Through the training process, instead of with real human players, de Vries et al. (2016) implemented a simulated oracle using a simple neural network (a combination of LSTM model and softmax layer), which attempts to output a final answer by minimising the *cross-entropy* error. On the other hand, instead of outputting yes/no answers, the guesser/questioner, implemented with a combination of LSTM and Recurrent Neural Networks (LSTM+RNNs), generates a sequence of relevant questions to handle a long-term conversational context.

Following the descriptions of previous work above, the online learning procedure, which we investigate here, is clearly more appropriate for multi-modal systems or robots that are expected to continuously, and incrementally learn from the environment and their users. In

the following section, we will discuss how others handle natural conversations with human tutors (either using rule-based or optimised dialogue strategies) in support of implementing a multi-modal teachable system.

2.2.3 Hand-Crafted versus Optimised Dialogue Strategy

Following the discussion in the previous sections, there has recently been previous work that shows increasing interest and good progress on addressing the language grounding problem through conversation with humans or other agents. Such multi-modal, interactive systems that involve grounded language are either: (1) *rule-based* as in e.g. Schlangen (2016), Skocaj et al. (2016), Tellex et al. (2013), Thomason et al. (2016a, 2015), Yu et al. (2016b): in such systems, the dialogue control policy is hand-crafted, and therefore these systems are *static*, cannot adapt, and are less robust; or (2) *optimised* as in e.g. Das et al. (2017), Strub et al. (2017), Whitney et al. (2017), Yu et al. (2016c): in contrast such systems are learned from data, and live interaction with their users; they can thus *adapt* their behaviour dynamically not only to particular dialogue histories, but also to the specific information they have in another modality (e.g. a particular image or video).

Strub et al. (2017) make an extension of their previous work (de Vries et al., 2016) by incorporating Deep Reinforcement Learning (DRL) with policy gradient approaches to train a question-generator (QGen) for the GuessWhat?! game. They train a question generation policy by letting the pre-trained oracle and guesser models interact with each other. Different to typical RL methods for a dialogue policy, which learns to optimise the procedure/sequence of actions to achieve a final goal, the QGen is trained to produce a sequence of words for specific question, and also learned to optimise the sequence of these questions through interaction with humans. The reward function for training the QGen agent depends on the success of the guesser's prediction (i.e. "if the correct object/item word is found in the generated questions, return 1, otherwise 0") (Strub et al., 2017).

Similar to Strub et al. (2017)'s work, Das et al. (2017) also extend their previous work (Das et al., 2016) to implement a pair of "cooperative visual dialogue agents", who perform as the questioner (Q-BOT) and the answerer (A-BOT) to attempt to find the correct image that the A-BOT desired through interaction with each other on the target scene (image). Both agents are trained to process Natural Language conversations using a DRL model. This work attempts to learn the generation of suitable question-answer pairs from both agents to improve the performance of image estimation. In order to keep the generality between two dialogue bots in DRL training settings, Das et al. (2017) introduced a "meta-agent", which is comprised of two "situated agents". However, due to different conversational roles, there are still several differences in the model settings between the two agents: for instance, 1)

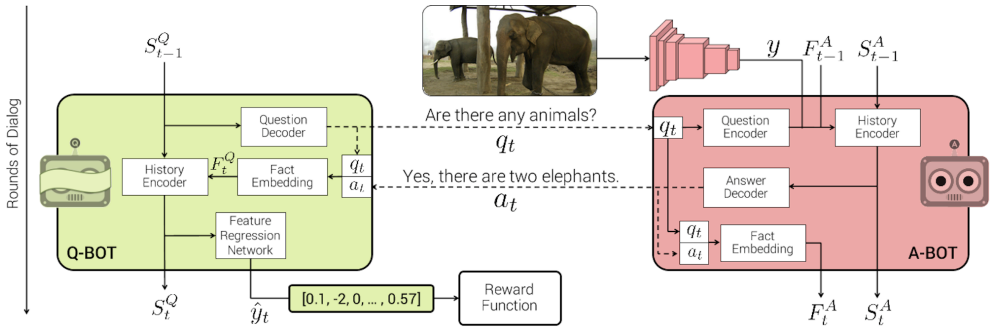


FIGURE 2.8: Policy networks for Q-BOT and A-BOT (Das et al., 2017)

compared to the A-BOT, Q-BOT contains an infinite action space with a large number of questions, where two agents are given different lexicon set; and 2) in terms of the information sharing, the A-BOT always takes the current image as part of its state space, but the Q-BOT cannot. In addition, the DRL model sets a local reward function for each state-action pair, where a distance metric between the estimated image representation and the target representation is applied to penalise the pair of question-answer, which lead to a worse prediction of the target image representation (Das et al., 2017). Meanwhile, it also provides a global reward for awarding/penalising the time steps that Q-BOT applied to improve the feature prediction. For synchronously training dialogue policies for both agents, Das et al. (2017) designed a policy network (Fig. 2.8), in which both agents policies are modelled using a "Hierarchical LSTM-based Recurrent Encoder-Decoder neural network" (see more details from Das et al. (2017)).

Whitney et al. (2017), which is more close to what we are trying to address using RL in this thesis, attempts to handle uncertainty (including noise in both speech and gesture observations) in the field of human-robot collaboration. Different with their previous work (Tellex et al., 2014, 2013), which repeatedly completed the specific task (i.e. moving the specific pallet onto the trunk) following Natural Language instructions from humans, here Whitney et al. (2017) attempts to reduce the mistakes from social feedback by interacting with human partners. In order to address this problem, Whitney et al. (2017) proposed a novel Partially Observable Markov Decision Processes (POMDP) in the item-fetching domain – FEedback-To-Collaborative-Handoff POMDP (FETCH-POMDP) (see Fig. 2.9) – that aims to learn to ask clarification questions (for instance, "Human: can I have the marker? Robot: this one?") when confused. The FETCH-POMDP model is implemented with a global reward function (which depends on whether the agent can pick up the human's desired item); a set of actions on both social feedback and physical aspects; as well as a set of observation models which aims at the selection probabilities on each possible item i given language l and gesture g from a human player.

Different to Whitney et al. (2017), the uncertainty we consider in this thesis results from

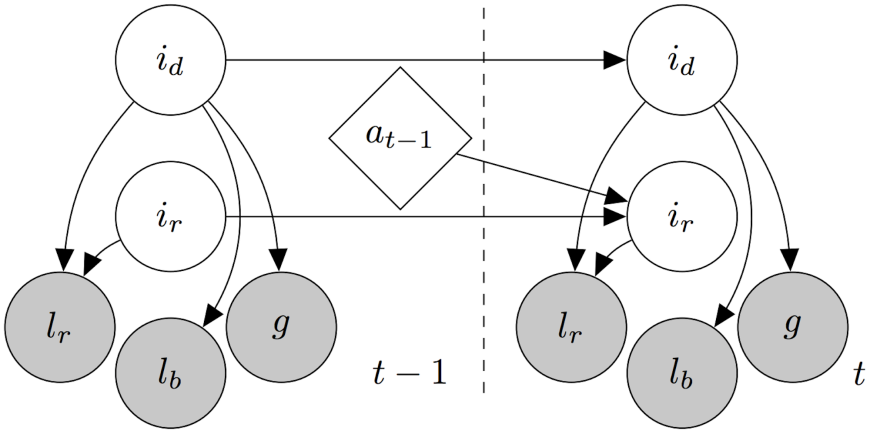


FIGURE 2.9: Architecture of Fetch-POMDP (hidden variable are coloured in white, observed ones are in grey) (Whitney et al., 2017)

the perceptual classification (visual classifier confidence scores). On the other hand, instead of processing the uncertainty under a constant environment, here what we are facing is a dynamic variable learning environment, which leads to more challenge in effectively learning novel knowledge through human feedback. As discussed in Chapter 1, the agent in this thesis does not have any basic knowledge about the visual scene (without visual classifiers). In the beginning of the learning process, the scores from classifiers are meaningless and unstable, but after training an increasing number of visual examples, the visual/external environment will generate more reliable classifier results. In order to cope with this challenge, we set up a separate MDP layer to resolve the uncertainty within such environment (see more details in Chapter 8).

Ideally, such interactive systems ought to be able to handle natural, spontaneous human dialogue. However, most work on interactive language grounding learn their systems from synthetic, hand-made dialogues or simulations which lack both in variation and the kinds of dialogue phenomena that occur in everyday conversation; they thus lead to systems which are not robust and cannot handle everyday conversation (Skocaj et al., 2016, Yu et al., 2016c,d). In this work, we try to overcome this limitation by training an adaptive learning agent from *human-human dialogues in a visual attribute learning task*.

2.3 Teachable Systems

This section describes and discusses some existing interactive systems (Kimura et al., 2013, Skocaj et al., 2016) that learn visual objects and properties through interaction with human tutors, as detailed below.

2.3.1 SOINN-Robots

Kimura et al. (2013) has demonstrated an interactive learning systems based on the SOINN models (described in Chapter 4), so-called SOINN-Robot. It is able to learn objects and their perceptual attributes through an interaction with the external environment via multiple sensors, e.g. camera and depth, 3D-information, and weight. This system deploys a pattern-based multi-layer architecture (see Fig. 2.10) (Kimura et al., 2013). It extracts low-level features from different sensor modalities on the "*Input*" layer. The proposed system copes with each sensor modality separately with an equal importance, rather than that of using a combination of certain modalities (Kimura et al., 2013). The extracted features are clustered into patterns with N dimensional data-points using a novel learning model (STAR-SOINN by Kimura et al. (2013)) on the "Pattern" layer. Eventually, these clustered patters are applied to learn a binary classifier for each labelled attribute, where the output of classifiers, like confidence values, are used to represent how strongly an attribute can be associated with a modality. The system is therefore able to learn/describe a particular attribute by properly processing one or more specific modalities. On the other hand, these confidence values are also stored into an object-attribute dictionary for the future learning and identification on the "Symbol" layer.

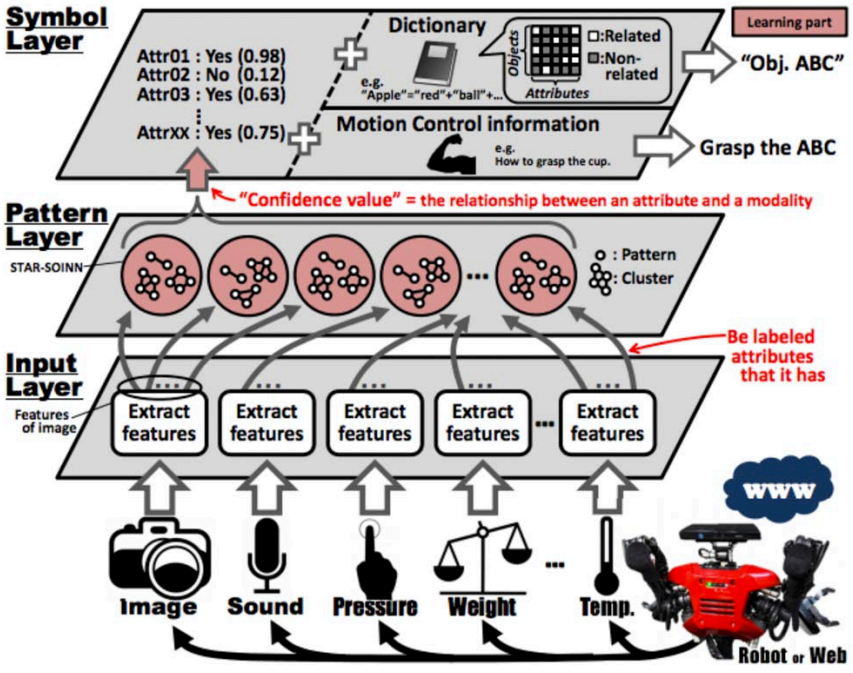


FIGURE 2.10: Architecture of the Pattern-based SOINN-Robot System (Kimura et al., 2013)

The SOINN-Robot demonstrates an ultra-fast, multi-modal, and incremental transfer learning method that allows guessing "unknown" objects by transferring the learned knowledge

from other objects in memory (Kimura et al., 2013). Furthermore, this robot is also provided with several robust self-processing mechanisms, as below:

- ***Self-Learning***: the system can learn and reproduce how humans behave on a particular task by itself (Kimura et al., 2013). It captures all potential and relevant features through diverse sensors and keeps estimations with clustered feature patterns in memory. Moreover, it supports learning the "unknown" knowledge by transferring online resources from other learning systems/robots while nothing happens (Kimura et al., 2013).
- ***Self-Guessing***: the system is able to make an accurate guess on unseen objects using its past experience and relevant knowledge in its own memory (Kimura et al., 2013). For instance, the robot maybe describe an unknown paper cup as "I guess it has attributes of cylinder, paper-made, soft and etc.".
- ***Self-Thinking and Acting***: Given a specific task request (for instance, "Please pour water into the cup."), the robot is able to make a decision about what actions need to be taken and also think about the potential sequence of these actions in practice (Kimura et al., 2013), based on the detected environment, like "I will pick up the blue bottle" or "I will get this glass".

Hence, different with the conventional industrial/experimental robots that perform well only on specific tasks, the SOINN-Robot can still achieve a higher completion and success rate for requested tasks by applying basic knowledge into immediate situation, even if the external environment has slightly changed (Kimura et al., 2013). For example, the robot is request to pour the water into two different coloured cups.

However, as research of the proposed SOINN-Robot mainly focuses on how a robot can efficiently learn to accomplish specific tasks using multi-modal information in the field of Machine Learning, it does not involve a NL interaction with humans for teaching, although the robot has been equipped with a microphone for capturing commands/descriptions. Our research in this thesis therefore is more concerned with a human-like dialogue interaction in for teaching grounded language.

2.3.2 The George System

Skocaj et al. [2011, 2016] present a fully integrated interaction system – the George System – that, different with the SOINN-Robot, focuses on communicating and learning the categorical knowledge through NL dialogue with the human tutors. This robot is able to

learn and refine the conceptual models of visual objects and corresponding properties, by receiving information from the tutor (e.g. "T: This is a Coke can"), or by taking the initiative by itself by asking questions (e.g. "L: Is this elongated object yellow?") (Skocaj et al., 2011, 2016). This research mainly focuses on 1) detecting knowledge gaps using a statistical framework and a curiosity-driven goal formation across multiple modalities; and 2) properly integrating individual solutions for different purposes, i.e. visual perception models and processing of linguistic context via a set of beliefs representing the states of the external world. These beliefs are viewed as intermediates that are formed for containing the representations of information from different modalities in a learning process Skocaj et al. (2016) and updated for the current state from the external world. The George robot is designed based on a distributed asynchronous framework Skocaj et al. (2016) (see Fig. 2.11), which can facilitate diverse components into the system in a comprehensive way.

The robot mainly contains two essential sub-architectures (SA) equipped with multiple components:

1. ***Visual SA*** for learning and recognising visual object properties Skocaj et al. (2016) using multidimensional features (e.g. a multiple 1D vector relating to colour, shape and texture of the observed objects) and the oKDE incremental learning models introduced in Chapter 4;

2. ***Dialogue SA*** that produces situated dialogues in an task-oriented interaction with humans, where the system may *understand* what the tutor intends it to do with specific knowledge/information within a large joint activity context (Skocaj et al., 2016). It applies a continual abduction proposed by Janícek (2011) for producing and verifying hypotheses of the human behaviours regarding to communicative intentions.

On the other hand, one of the essential contributions of the George system is to interactively learn novel object properties under multiple learning strategies driven by either the tutor or the system itself. It is more concerned with the tutor-initiative interaction, where human tutors always drive the dialogue forward in a learning process. It defines and compares different learning strategies based on different tutor's behaviours:

- ***Situated tutor-driven strategy*** shows a specific learning situation, in which the tutor may explicitly teach the robot knowledge about particular objects (Skocaj et al., 2016). While information provided by a tutor is successfully assigned to an object detected by the visual SA, a sequence of learning actions will be executed for each property by the tutor to update both internal visual models (the oKDE model) and corresponding status beliefs.

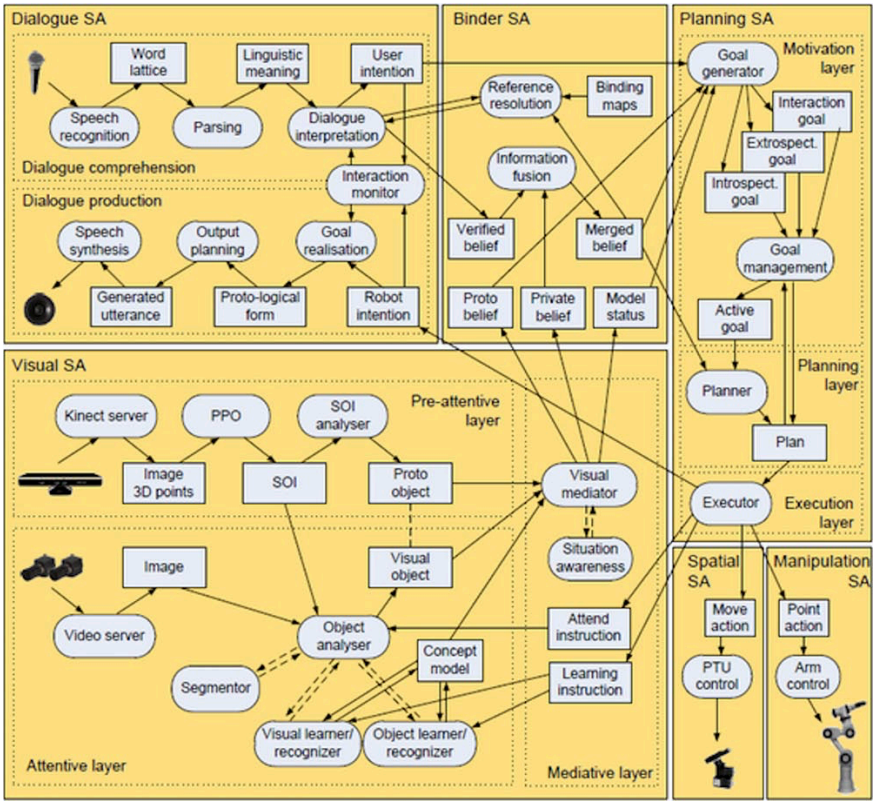


FIGURE 2.11: Architecture of the George System (Skocaj et al., 2016)

- ***Situated autonomous strategy*** addresses a learning situation where the system executes an autonomous learning cycle with updating an internal visual model while information provided by the visual SA itself is reliable (i.e. the visual concepts are identified with a high confidence score) (Skocaj et al., 2016). Otherwise, the system will describe the specific object with uncertainty, like "learner: this is probably a coke can.". However, unfortunately, Skocaj et al. pointed out that this strategy may take a risk of incorporating with incorrect recognised information, where sometimes a high confidence score may present a false impression about the reliability of its identifications.

- ***Situated tutor-assisted strategy***, in contrast to the tutor-driven strategy, the tutor may passively wait for queries from the system rather than actively provide available information about particular objects. This strategy relies on the recognition ability of the system, who can ask polar questions (e.g. "learner: is this a ") or request conformations (e.g. "learner: what colour is this coke can?") about a object's attributes (Skocaj et al., 2016). Skocaj et al. also emphasised that, after the tutor answers, the system will take a sequence of learning actions, similar to the tutor-driven learning.

Similar to Skocaj et al.'s work, we are also concerned with comparison of different dialogue strategies in an interactive learning process. However, different with their work, we intend

to investigate effectiveness of diverse task-oriented dialogue behaviours, from both sides of the tutor and the learner/system, on the final learning performance, and then explore a more appropriate mechanism for learning object properties in a natural way (as detailed in Chapter 7).

Furthermore, the George system also proposed a non-situated tutor-assisted learning that allows the system to make a request on more examples for specific object attributes (Skocaj et al., 2016), e.g. "learner: can you show me something red?". This strategy is based on an introspective learning mechanism, where the system may make a judgement about what needs to be requested based on previous learning performance (e.g. accuracy or other recognition measures). *Note* that this introspective learning strategy is deployed with the lowest priority, which means it is only executed while nothing else is happening.

2.4 Dialogue Processing

As discussed above, in order to learn novel visual concepts (e.g. colour and shape) within more complicated situations, communicating with humans should provide positive impacts on an overall learning performance (or the quality of symbol grounding). For processing a natural conversation with human beings, dialogue processing is considered as an important component given the interactive visual-concept learning task. In this section, we will briefly describe basic models of dialogue systems, and also discuss related approaches, and data as well as user simulation models.

2.4.1 Standard Spoken Dialogue Systems

A Spoken Dialogue System is defined as a computer-based system that can support communication between humans and machines through spoken language (Heinroth and Minker, 2012, Lison, 2014). Figure 2.12 presents an overview architecture of a typical spoken dialogue system, which is built with a chain of processes, consisting of three essential layers, as below:

Acoustic Front-end: this layer, normally accessed by microphones, is constituted by speech – the *automatic speech recognition (ASR)* and speech synthesis – *text-to-speech (TTR)* (Heinroth and Minker, 2012). ASR is applied to analyse and extract features from the audio signal and then transforms it into a textual hypothesis of the utterance. TTR synthesis is applied in the opposite direction of the ASR, in which it aims to generate the audio to the human user (Skantze, 2007).

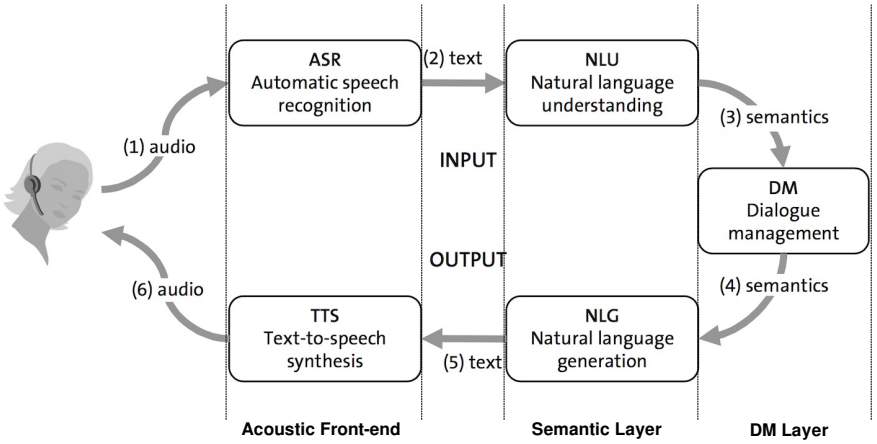


FIGURE 2.12: Standard Architecture of a Spoken Dialogue System (Heinroth and Minker, 2012, Skantze, 2007)

Semantic Layer: the semantic layer, consisting of two modules (*natural language understanding (NLU)* and *natural language generation (NLG)*), is applied as an indispensable bridge between the other two layers. Specifically, the NLU module (or linguistic analyser) is designed to computationally parse the output of the speech recogniser by mapping it into a particular semantic representation (Skantze, 2007). These semantic representations will be analysed and processed by the last layer (Dialogue Management Layer). Reversely, the NLG module (or text generator) needs to interpret semantic representations to generate a surface representation of the utterance to the user (Skantze, 2007). Given the domain of visual grounding, the semantic layer plays an essential part in retrieving and grounding the visual concepts through spontaneous conversation with human tutors (more details are shown in Section 2.4.3)

Dialogue Management (DM) Layer: the DM layer (Heinroth and Minker, 2012), as the final but important layer in the process chain, deploys *dialogue management*, which generally links the semantic representation output from the NLU module with the one interpreted by the NLG. In detail, given a particular semantic representation, the DM will look at the discourse as well as the previous dialogue context to determine what dialogue action (e.g. ask or answer questions, make statements, ask for clarification, as well as acknowledge/reject the previous utterance) will be taken next, which will generate a response to the user on a semantic level (Kennington, 2016, Skantze, 2007). The dialogue manager can select the next action by 1) following a set of pre-defined rules, or 2) learned or planned dialogue strategies using e.g. Reinforcement Learning or Neural Networks (e.g. Sequence-to-Sequence model) – both of which we will apply to implement the dialogue manager for specific purposes in this thesis (see more in Chapter 3).

2.4.2 Situated Dialogue Systems

Following the definition by Lansdale and Ormerod (1994), situated dialogue is viewed as "a joint process of communication", in which information (for instance, "data, symbols as well as context") may be shared between multiple parties. In the context of the grounding problem through human-robot interaction, Kennington (2016) specified that, different to a classical dialogue system, which usually considers speech as the unique modality for the interaction between human and machines, through a situated dialogue system, participants (including human beings and the system) can see symbols, objects, or gestures in a shared environment (called World (W)), which enables the system to "observe the non-linguistic but communicative context from its interlocutor" (see Fig. 2.13).

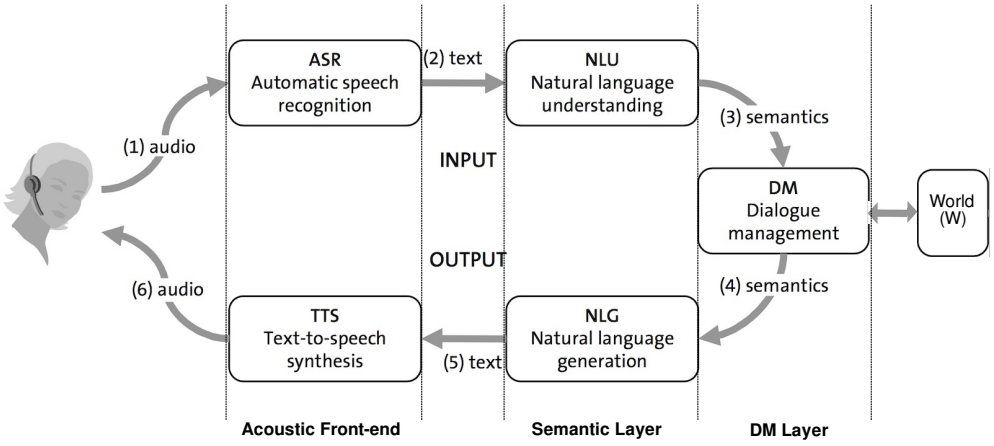


FIGURE 2.13: Loosely Following the Architecture of the Situated Spoken Dialogue System from Kennington (2016)

Given the visual language grounding task in this thesis, W will represent individual visual objects (called visual/non-linguistic context) presented in the shared space (we will explain how the representation of W can be generated and also how this representation can be grounded through dialogue in later chapters.) In order to distinguish different individual objects from each other, a set of low-level feature information (for instance, raw or an extracted bag of visual features) will also be represented in W.

Situated dialogue systems play an essential role on resolving problems in which a system is required to be aware of the immediate environment/situation while interacting with human beings over time. For instance, in the context of human-robot interaction or collaboration (Chai et al., 2016, Kennington and Schlangen, 2014, Kollar et al., 2013, Tellex et al., 2013), when the robot speaks with human users, since the user knows everything within the certain situation (e.g. task-goal and potential task -constraints or -limitations), the user can find out and adopt the best strategies that can guide the system to perform correct actions to efficiently achieve the final goal.

The interaction with humans described above is considered as a situated dialogue. However, regarding more complicated human-robot tasks with many variations and uncertainties, human beings cannot always present a clear mind, especially in the very beginning of their speaking, which usually involves a diversity of incremental dialogue phenomena, such as hesitation, self -repetition and -repair (which require incremental dialogue processing, which is discussed in the following section.

2.4.3 Incremental Dialogue Systems

Standard dialogue systems usually assume that the conversation with human beings proceeds turn-by-turn (Allen et al., 2001, Howes et al., 2011). However, different to such assumptions, everyday human conversation contains natural, spontaneous speech, which is highly interactive and more complicated, with many interruptions and overlaps (see dialogue (a) in Table. 2.1). In addition, since natural, spontaneous dialogue is inherently incremental (Crocker et al., 2000, Ferreira, 1996, Purver et al., 2009a), it gives rise to dialogue phenomena such as *self-* and *other-corrections*, *continuations*, *unfinished sentences*, *interruptions*, *hedges*, *pauses* and *fillers* (see example (b) in Table. 2.1).

L: this colour is ... [red], red T: [red]. T: good. and the shape is ... L: square? T: yes it is.	L: erm... this is green, right? T: no, it is purple sorry blue, blue triangle. L: triangle? T: yes, well done.
(a) overlap & continuation	(b) self- correction & repetition

TABLE 2.1: Natural, Incremental Dialogue Examples (T: tutor, L: learner) for the Interactive Grounding Task

In order to address this problem, Incremental Dialogue Processing (IDP) was introduced to extend the typical dialogue system by modelling real human behaviours in such natural conversations, where humans can process the utterances word by word, and meanwhile determine the most appropriate response, barging in if necessary (Ghigi et al., 2014, Schlangen and Skantze, 2009). Ghigi et al. (2014) indicate that, given task-oriented conversations, a number of long, complicated utterances (with dialogue phenomena, e.g. self-correction) are usually difficult for the system to correctly understand, which result in worse responses, bad user experiences, and even task failures. In such cases, one of main purposes of IDP is to assist the dialogue system with achieving a relative balance between user experience and task success as well as correct input: minimising the user annoyance and incorrect output while maximising the correct input.

In the past few years, there has been a surge of significant progress on improving the quality of incremental dialogue processing in speech recognition (Kennington et al., 2014,

Schalkwyk et al., 2010, Walker et al., 2004) and synthesis (Buschmeier and Kopp, 2013, Skantze and Hjalmarsson, 2010), and dialogue management (Buß et al., 2010, Selfridge et al., 2012). Moreover, Schlangen and Skantze (2009, 2011) have proposed an incremental model in support of building incremental dialogue systems. This incremental model, named the Incremental Unit (IU) framework, is an abstract architecture consisting of a network of processing modules, in which each module contains a *left* buffer, a processor as well as a *right* buffer. "The incremental unit (IU) is a basic unit of information which is communicated between the modules." (Schlangen and Skantze, 2011) The module typically takes a graph of IUs as input from the *left* buffer, executes some type of processing on the data, and outputs them from the *right* buffer.

In further work, Buß and Schlangen (2011) designed an incremental dialogue manager (DIUM) by incorporating the IUs from Schlangen and Skantze (2009, 2011), which, given a sequence of IU states representing semantic, discourse and dialogue action, manages *when* and *how* to revoke the decision/action that has been made previously based on a new intent the user desired. This DM model shows good performance on processing different types of corrections through real-time interaction with human users.

Schlangen and Skantze (2009, 2011)'s work provides a robust, abstract framework for designing and implementing incremental dialogue systems. It resolves the incremental processing problem via adding or revoking the incremental units in the middle of conversations. However, the framework is abstract, and does not possess the capacity of processing spontaneous dialogue with complex incremental dialogue phenomena (for instance, overlap, self-correction and continuation) on the semantic level. In order to handle such issues, some recent work from Hough (2014), Kennington and Schlangen (2014) incorporates the IU framework with different incremental NLU models. More specifically, Kennington and Schlangen (2014) deploys RMAS (a framework by Copestake (2007) "for representing semantics that factors a logical form into elementary predicates") to process a corresponding, underspecified semantic representation for each word increment. On the other hand, Hough (2014) deploys Dynamic Syntax (DS), which is a word-by-word incremental semantic parser/generator, based around the DS grammar framework Cann et al. (2005b) especially suited to the fragmentary and highly contextual nature of dialogue. In DS, dialogue is modelled as the interactive and incremental construction of contextual and semantic representations Eshghi et al. (2015). The contextual representations employed by DS are of the fine-grained semantic content that is jointly negotiated/agreed upon by the interlocutors, as a result of processing questions and answers, clarification requests, corrections, acceptances, etc. DS has recently been extended by incorporating with Type Theory with Records (TTR) – the formalism for DS contextual/semantic representations (see more details about the DS-TTR model in Chapter 3 and Appendix A)

In this thesis, similar to Hough (2014)'s work, we also approach incremental natural language understanding (NLU) using the DyLan incremental parser Eshghi et al. (2011) based on the DS-TTR formalism [3]. The distinctions with his work are that 1) instead of working on the semantic representation itself, we extend the DyLan parser to infer dialogue acts based on the completed semantic sub-trees from DS and also the corresponding maximal TTR representations; 2) instead of synthetic conversations, we deploy the DyLan parser to cope with realistic human-human conversations (see more details in Chapter 9).

2.5 Chapter Summary

Previous work	Compositional	Interactive (NL Conversation)	Optimised	Attribute Grounding	Natural Language	Incremental Learning	Semantic Representaion
(Kollar et al., 2013)	✓				✓		✓
(Tellex et al., 2013)	✓	✓			✓		✓
(Kimura et al., 2013)	✓			✓		✓	
(Matuszek et al., 2014, 2012)	✓	✓		✓	✓		✓
(Tellex et al., 2014)	✓	✓	✓		✓		✓
(Silberer and Lapata, 2014)					✓		
(Socher et al., 2014)					✓		
(Karpathy and Li, 2015)					✓		
(Kennington and Schlangen, 2015)	✓			✓	✓		
(Thomason et al., 2016a, 2015)	✓	✓		✓	✓		✓
(Das et al., 2016)		✓			✓		
(de Vries et al., 2016)		✓			✓		
(Schlangen, 2016)	✓	✓			✓		✓
(Skocaj et al., 2016)	✓	✓		✓	✓	✓	
(Strub et al., 2017)		✓	✓		✓		
(Das et al., 2017)		✓	✓		✓		
(Whitney et al., 2017)	✓	✓	✓		✓		✓
(Steels et al.)	✓	✓		✓		✓	

TABLE 2.2: Overview of previous work for addressing the visual language grounding problem

In this chapter, we have reviewed literature relevant to addressing the visual language grounding task through interaction with human beings. It mainly consists of two parts: 1) a review of the previous work that has approached the grounding task (including sections 2.1, 2.2 and 2.3), and 2) a summary of a standard situated dialogue system architecture followed by a discussion of their suitability for processing incrementality in natural, spontaneous conversations with human users.

In terms of the first part, we briefly introduced the symbol grounding problem (including a variety of definitions and theories of the grounding problem in AI and cognitive science), in which Harnad (1999), Steels (2008) also emphasised that, with more complicated situations, communication/feedback from humans can play an essential role in the symbol grounding problem for robots. Then we surveyed previous projects and also two existing teachable multi-modal systems for visually grounding word meanings.

[3] Unfortunately, we do not deploy the entire incremental units framework for building the dialogue module, because this thesis mainly concerns the interactive visual grounding problem, rather than incremental dialogue processing itself.

Following the motivation in chapter 1 and this survey of the literature, we argue that to effectively address the grounding problem, the learning framework/model needs to be compositional, trainable (using optimised strategies), and able to incrementally learn low-level visual knowledge through natural language conversations with humans. Here, we compared previous work regarding which of these properties they have dealt with for visual grounding and how they approach them

Table 2.2 shows a high-level comparison between existing systems or models across these properties, and indicates that, although there has been much research work showing good performance on grounding NL symbols into perceptual scenes, most research only considers one or some of these properties in their design and implementation. Different to all these projects, in the thesis, we believe that only a system which gives consideration to all these properties, can be robust and practical for real-world applications.

At the beginning of this project, we attempted to rebuild and extend the existing approaches/models described above to fulfil all properties. However, since most of models were designed for different research purposes/domains, they cannot be simply extended into a new domain we observed in this thesis by modifying one of their components, for example, the model by Kennington and Schlangen (2015) for reference resolution, Das et al. (2016), de Vries et al. (2016) for QA interaction about complex visual scenes, as well as Tellex et al. (2013), Whitney et al. (2017) for semantic grounding between robot movement with Natural Language instructions. On the other hand, some previous work, e.g. George system by Skocaj et al. (2016), Thomason et al. (2016b), are not fully available for the public, it leads to difficulties of reproducing their approaches or results. Hence, in this thesis, we propose a new framework from scratch, in support of building a multi-modal learning agent, which deploys trainable and *optimised* dialogue strategies to learn unknown visual *attributes* through *natural, spontaneous conversations* with human tutors *incrementally*, over time. Also the system should be able to learn these low-level features *on the fly* within a *dynamic* environment.

On the other hand, in terms of the interactive capacity of a teachable system for understanding what humans are saying in everyday incremental conversations, we have briefly reviewed different types of dialogue processing systems with relevant features and approaches, and also discussed their suitability and importance for addressing the visual language grounding problem: properly processing incremental dialogue phenomena (especially "*self-repair*") plays an essential role in effectively learning novel visual concepts through everyday incremental conversations with human beings. In the next Chapter, we will propose an interactive multi-modal framework, in which the vision and dialogue modules interact with each other via either dialogue act representations or formal semantics (TTR record types). This framework will be applied for the interactive visual language grounding task in this thesis, with minor updates for specific research purposes.

On the other hand, apart from the related work of the Interactive Language Grounding itself, there are also a list of essential technologies and work corresponding to different system's components or sub-tasks. We, therefore, will review those approaches in the subsequent chapters, for example, Reinforcement Learning, SARSA Algorithm, and introduce the Dynamic Syntax and Type Theory with Records (DS-TTR) model in Chapter 3; existing visual classification approaches for the visual attribute learning task (Chapter 4); existing human conversation corpora in the filed of multi-modal learning (Chapter 5); as well as existing user simulation models and their relevant approaches in Chapter 6. We will go through each of them with more details in the rest of the thesis.

Chapter 3

Multi-modal Framework for Interactive Language Grounding

Following the discussion in the previous chapter, a system for the interactive language grounding task should be capable of: (1) *compositional* grounding; (2) *incremental* learning of *visual-attribute* concepts; through (3) *natural, spontaneous dialogue* with humans; from (4) *as little data as possible*. To this end, in this chapter we introduce a modular **Interactive Multi-modal Framework** (as shown in Figure 3.1) in support of building such a teachable system, which gives consideration to all these properties. The proposed framework consists of two core modules: (a) a ***vision module*** in charge of extracting relevant low-level visual features, and visual classification (the real-time output from the vision module can be used to construct the visual, non-linguistic context in dialogue, providing for example the antecedents for deictic, and other context-dependent expressions) and (b) a ***dialogue module***, consisting of three core components: Natural Language understanding (NLU), generation (NLG) and dialogue management (DM)), which is designed for handling NL conversation with human tutors. For addressing the grounding problem within dialogue, the agent needs to: 1) *understand* what aspects of the visual environment the tutor is referring to and 2) *learn to naturally describe* its visual surroundings (e.g. visual objects or low-level attributes). In order to address both issues, the proposed framework incorporates representations that enable knowledge sharing and interaction between different modalities, for instance vision and language. It also incorporates mechanisms for processing natural, spontaneous dialogue with human partners.

Furthermore, given such an interactive learning task, it is not desirable for an intelligent system/robot to just simply replicate human behaviour, i.e. gaining visual knowledge by repeatedly asking human tutors same/similar questions through dialogue (e.g. "what colour is this?" and "what shape is this?'). We, therefore, build a dialogue management model with

optimised strategies trained using Reinforcement Learning (RL), where the agent learns not only to manage natural, human-like dialogue for the learning task, but also to perform a form of *active learning* that helps the agent determine when to ask for further information: the agent will ask for feedback or further help (e.g. clarification or confirmation) from real humans only when it is necessary, thus minimising human involvement/effort in the learning task.

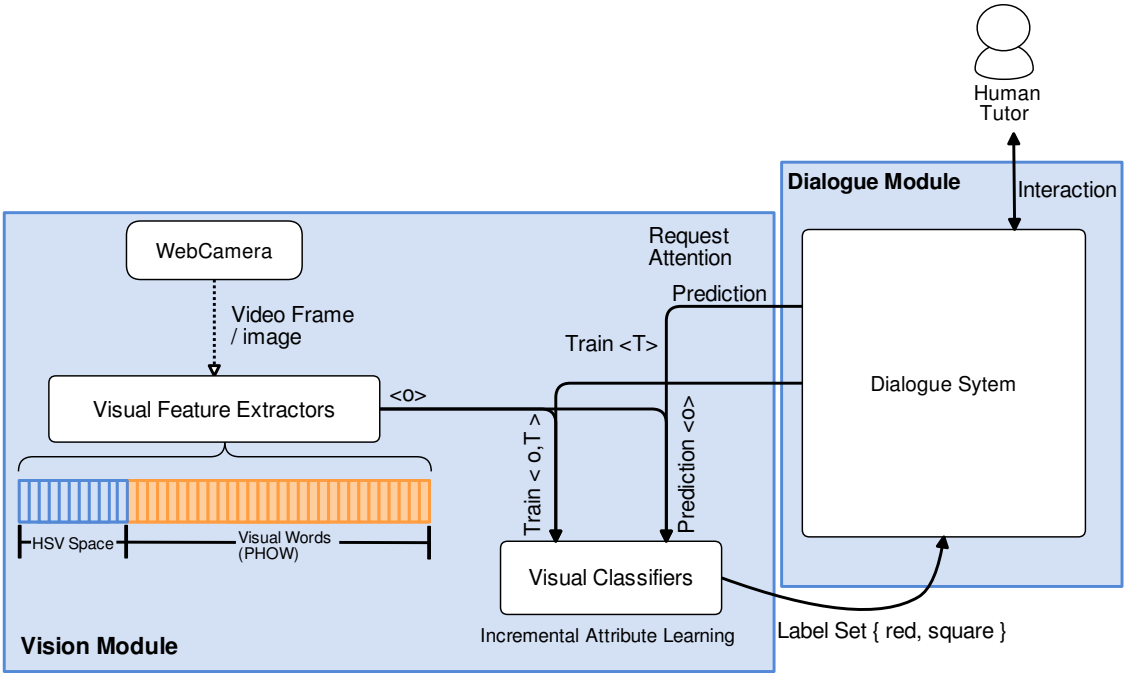


FIGURE 3.1: Architecture of the teachable system

More specifically, the contributions of work in this chapter are presented across the following sections:

- Section 3.1 introduces the vision module mainly from two aspects: 1) how visual features of an object is represented as well as 2) what classification models can be applied to incrementally learn and predict novel objects in an interactive learning process.
- Section 3.2 presents a standard dialogue architecture, consisting of NLU, NLG and DM components. We will present two potential understanding models: a SimpleSLU (hand-crafted dialogue action tagging) and a DS-TTR (incremental semantic parser). We also briefly present DM approaches, via either hand-crafted rules or optimised strategies using Reinforcement Learning.
- Section 3.3 further describe a simulated learning environment for training and testing the dialogue/learning strategies for the learner/agent. Within the environment, the

learner/agent can learn novel visual concepts by interacting with a simulated tutor built based on either synthetic or realistic dialogues.

- Section 10.1 raises a discussion on the justification of using either a simple hand-crafted parser (SimpleSLU) or a complex parser (DS-TTR) in the interactive learning task. Following this discussion, although the DS-TTR model is more complex and also needs more work on grammar and lexicon coverage, it plays an essential role in processing natural conversations given the learning task.

Finally, 3.4 will summarise what we achieved in this chapter.

3.1 Vision Module

The vision module is designed to cope with the visual learning issues in the field of computer vision that focuses on extracting, processing as well as identifying perceptual attributes, such as colour, shape and material. It involves two main sub-tasks, i.e. visual feature extraction (section 3.1.1) and attribute-based classification (section 3.1.2). This module applies several engineer-level extraction approaches to produce feature representations for specific visual objects once new object images or video frames are presented. It applies an incremental SVM method with a stochastic gradient descent policy (SGD-SVM) to learn or identify novel objects using such feature representations. To our knowledge, although a high-level performance of visual classification has a good chance to rely on the robust deep learning techniques (Donahue et al., 2015, Karpathy and Li, 2015, Kiros et al., 2014, Venugopalan et al., 2016, 2014, Vinyals et al., 2015), it normally requires a significant longer computational time than traditional machine learning approaches, and also a huge amount of training examples with the high-level quality, which are both unacceptable for a life-long learning task.

On the other hand, as this research focuses mainly on investigating the effects of NL dialogues on object-learning tasks, we can only explore and implement an appropriate framework in vision module, but not dive too deep into alternative techniques in the field of computer vision. Since the framework proposed is modular, we can easily replace the SGD-SVM models or feature extraction methods with more robust approaches to improve the overall performance once they are developed.

3.1.1 Feature Extraction

A feature representation consisting of visual attributes will be required for learning to classify and describe novel objects. In contrast with previous work by Farhadi et al. (2009),

to reduce the noise from visual features on the stage of feature extraction, the feature representation has been simplified with two base feature categories, i.e. the colour space for colour attributes, and a "bag of visual words" for the object shapes/class.

The colour descriptor, consisting of HSV colour space values, are extracted for each pixel and then are quantized to a $16 \times 4 \times 4$ HSV matrix. These descriptors inside the bounding box are binned into individual histograms. Meanwhile, a bag of visual words is built in PHOW descriptors using a visual dictionary, which was calculated using visual object collection (described in following Chapters). These visual words will be calculated using 2x2 blocks, a 4-pixel step size, and quantized into 1024 k-means centres.

The feature extractor in the vision module presents a 1280-dimensional feature vector for a single training/test instance by stacking all quantized features (see Fig. 3.1).

3.1.2 Attribute-based Classification

Given the learning task, attribute-based classification plays an important role on identifying and learning novel visual objects and relevant features. The proposed framework applies the extracted feature-vector for a particular object as input to predict possible attribute labels and corresponding confidence scores Here, we design the visual module and its interaction with the dialogue module in a general way. As such it can be integrated with most of the state-of-the-art classification approaches. Following our main aim of an on-line learning process mentioned above, the chosen method should undertake a relative balance between the recognition performance of the classifiers on the one hand and issues of time, amount of data needed, and incrementality of the learning models on the other. Hence, we expect an incremental learning method that learns novel knowledge from small amount of data. In this thesis, we compare a list of state-of-the-art methods that have previously shown good performance on image-labelling tasks, including multi-label classification, single-label offline and online learning models. We finally apply an incremental SVM model (logistic regression SVM with stochastic gradient descent (SGD-SVM)), which is a simple but efficient approach, to learn binary feature (e.g. redness or not, square or not) for each visual instance (see more details in Chapter 4). Similar to the previous work (Farhadi et al., 2010, Matuszek et al., 2014, Silberer et al., 2013), instead of grouping a set of similar visual words as a single attribute, we consider each word (for example, "red" and "square") as an attribute value and both "colour" and "shape" are attribute categorise, which support explicitly grounding each visual-attribute word/atomic item (from the formal semantic) to a particular visual classifier. Given the visual-attribute learning task in the Dialogue Collection (see Chapter 5), before learning specific visual-attribute words, the participant has already known their

categories (e.g. colour and shape), which is similar to our language model settings, i.e. the parser has predefined these categories within its language grammar and lexicon[1].

Our vision model is simply extended to learn new visual features, e.g. texture. In terms of the feature extraction, since we employed an engineer-level extractor method that separately detects different low-level features and restructures them in a high-dimensional feature representation, it allows us to scale to additional visual/non-functional features (e.g. texture). On the other hand, we train a set of binary classifiers, each for a single visual attribute, which also helps us easily and quickly learn new individual features by creating new classifiers, without affecting each other.

3.2 Dialogue Module

The dialogue module is in charge of processing and controlling the interaction with human tutors through NL dialogue. As such it has three main components: NL understanding (NLU), NL generation (NLG) and dialogue management (DM) as usual. The dialogue module interacts with the vision system in two ways: (1) the semantic representations (either Types in Type Theory, or Dialogue Acts, see below) produced & referenced in its NLU & NLG modules are grounded in the individual classifiers within the vision module, for example, for the words 'square' and 'red', in "this is a red square, right?" or "what is the colour of this square?"; and (2) the vision module provides the visual, non-linguistic, context of the dialogues, for e.g. the resolution of pronouns and definite descriptions.

As noted above, one of the main aims of this thesis has been to explore different methods for dialogue processing (e.g. incremental or not, grammar-based or not, etc.), and their implications for the interactive visual grounding task. Hence, we introduce below two alternative methods for NLU/NLG: (1) deep semantic processing using the Dylan parser: implementation of the Dynamic Syntax and Type Theory with Records formalism (DS-TTR); and (2) shallow Dialogue Act classification, and template-based generation for NLG.

This is then followed by a description of Dialogue Management (DM) methods. The DM component is either rule-based (Chapter 7) or optimised using Reinforcement Learning (Chapters 9 & 8). See below for more details.

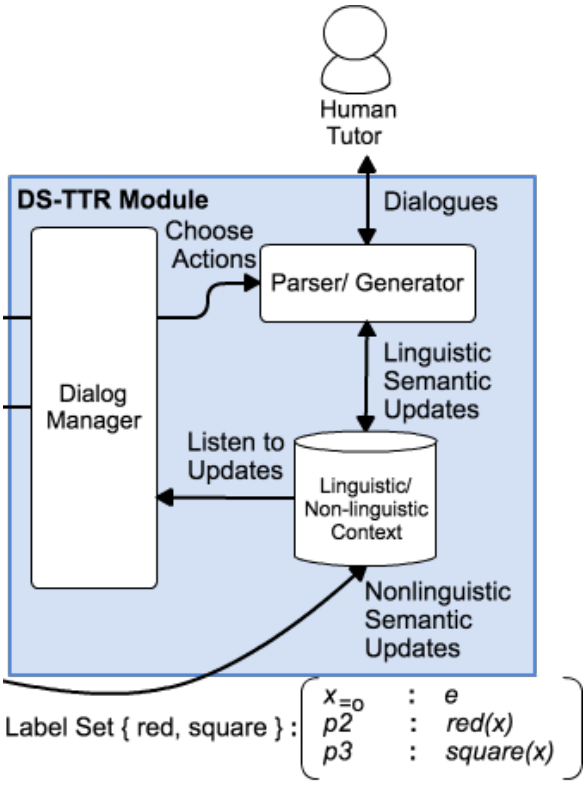


FIGURE 3.2: Architecture of Dialogue Module incorporating with the Dylan parser

3.2.1 Incremental Parsing and Generation: Dynamic Syntax and Type Theory with Records (DS-TTR)

In this section we briefly introduce and motivate the Dynamic Syntax (DS, Cann et al. (2005b), Kempson et al. (2001)) and Type Theory with Records (TTR, Cooper (2005), Cooper and Ginzburg (2015)) formalisms, in addition to how the DS-TTR parser, Dylan (Eshghi, 2015, Eshghi et al., 2011) is integrated into our overall interactive grounding framework.

3.2.1.1 Dynamic Syntax (DS)

Dynamic Syntax is a parsing-directed grammar formalism, which models the word-by-word incremental processing of linguistic input. Unlike many other formalisms, DS models the incremental building up of *interpretations* without presupposing or indeed recognising an independent level of syntactic processing. Thus, the output for any given string of words is a purely *semantic* tree representing its predicate-argument structure; tree nodes correspond to terms in the lambda calculus, decorated with labels expressing their semantic type (e.g.

[1] As part of the future direction, the agent/robot needs to jointly learn the certain visual attributes and their corresponding categories through dialogue (see more details in Chapter 10)

$Ty(e)$) and formula (Record Types or lambda abstracts in TTR, see the next section for details), with standard beta-reduction determining the type and formula at a mother node from those at its daughters (Figure 3.3).

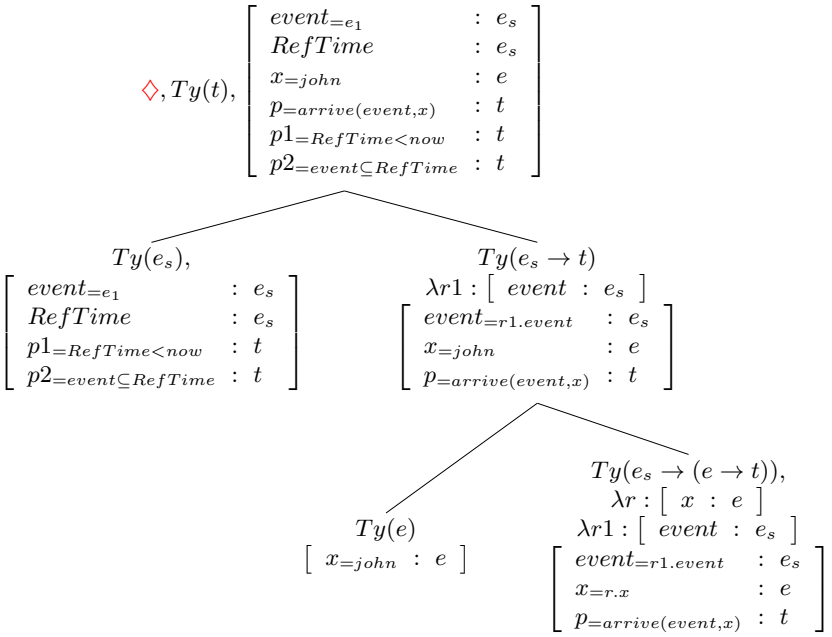


FIGURE 3.3: Semantic tree after parsing "John arrived"

These trees can be *partial*, containing unsatisfied requirements for node labels (e.g. $?Ty(e)$ is a requirement for future development to $Ty(e)$), and contain a *pointer* ◇ labelling the node currently under development. Grammaticality is defined as parsability: the successful incremental construction of a tree with no outstanding requirements (a *complete* tree) using all information given by the words in a sentence. Note that in these trees, leaf nodes do not necessarily correspond to words, and may not be in linear sentence order (see Figure 3.3); and syntactic structure is not explicitly represented, only the structure of semantic predicate-argument combination.

Actions & the parsing process The parsing process is defined in terms of conditional *actions*: procedural specifications for monotonic tree growth. These take the form both of general structure-building principles (*computational actions*), independent of any particular NL, and of language-specific actions induced by parsing particular lexical items (*lexical actions*).

These actions define the parsing process. Given a sequence of words $(w_1, w_2, ..., w_n)$, the parser starts from the *axiom* tree T_0 (a requirement $?Ty(t)$ to construct a complete tree of propositional type), and applies the corresponding lexical actions $(a_1, a_2, \ldots, a_n)$, optionally interspersing computational actions. Sato (2011) shows how this parsing process can be modelled on a *Directed Acyclic Graph* (DAG), rooted at T_0, with partial trees as nodes, and

computational and lexical actions as edges (i.e. transitions between trees). See Appendix A for a more thorough illustration of the parsing process in DS.

This parse search process is modelled as a Directed Acyclic Graph (DAG) in which nodes are partial semantic trees, and edges are either computational, or lexical actions (Eshghi et al., 2011, Sato, 2011). Later work has shown how this DAG constitutes the linguistic context of the conversation (Eshghi et al., 2012, 2015, Kempson et al., 2015), used to model various types of fragment construal as well as self-corrections (see chapter 9), but also for representing the jointly agreed content of the conversation. Here, like in other places (e.g. Hough (2015)), we take a more coarse-grained view of the DAG with edges corresponding to sequences of computational actions followed by a single lexical action corresponding to the word parsed - see Figure 3.4 and Figure 3.4.

Dialogue Processing Given the inherent incrementality of Dynamic Syntax, it is especially well suited to the fragmentary and highly contextual nature of dialogue. In DS, dialogue is modelled as the interactive and incremental construction of contextual and semantic representations (Eshghi et al., 2015, Purver et al., 2011). The contextual representations afforded by DS are of the fine-grained semantic content that is jointly negotiated/agreed upon by the interlocutors, as a result of processing questions and answers, clarification requests, corrections, acceptances, etc. (see more details in Appendix A or Eshghi et al. (2015) for an account of how this can be achieved grammar-internally as a low-level semantic update process).

Generation (linearisation) in DS is defined using trial-and-error parsing, with the provision of *a generation goal*, viz. the semantics of the utterance to be generated. Generation thus proceeds, just as with parsing, on a word-by-word basis (see Appendix A and Hough (2015), Purver et al. (2014) for details).

The upshot of all this is that using DS, we can not only track the semantic content of some current turn as it is being constructed (parsed or generated) word-by-word, but also the context of the conversation as a whole, with the latter also encoding the grounded/agreed content of the conversation (see Eshghi et al. (2015), Purver et al. (2010) for details).

Recent work (Eshghi et al., 2012, Hough, 2011, Purver et al., 2014) has extended the Dynamic Syntax framework by incorporating Type Theory with Records (TTR) as the logical formalism in which meaning representations are couched (Eshghi et al., 2012, Purver et al., 2011) (see Figures 3.3 and 3.4 for example Record Types in TTR). We will now proceed to motivating and introducing TTR in the next section.

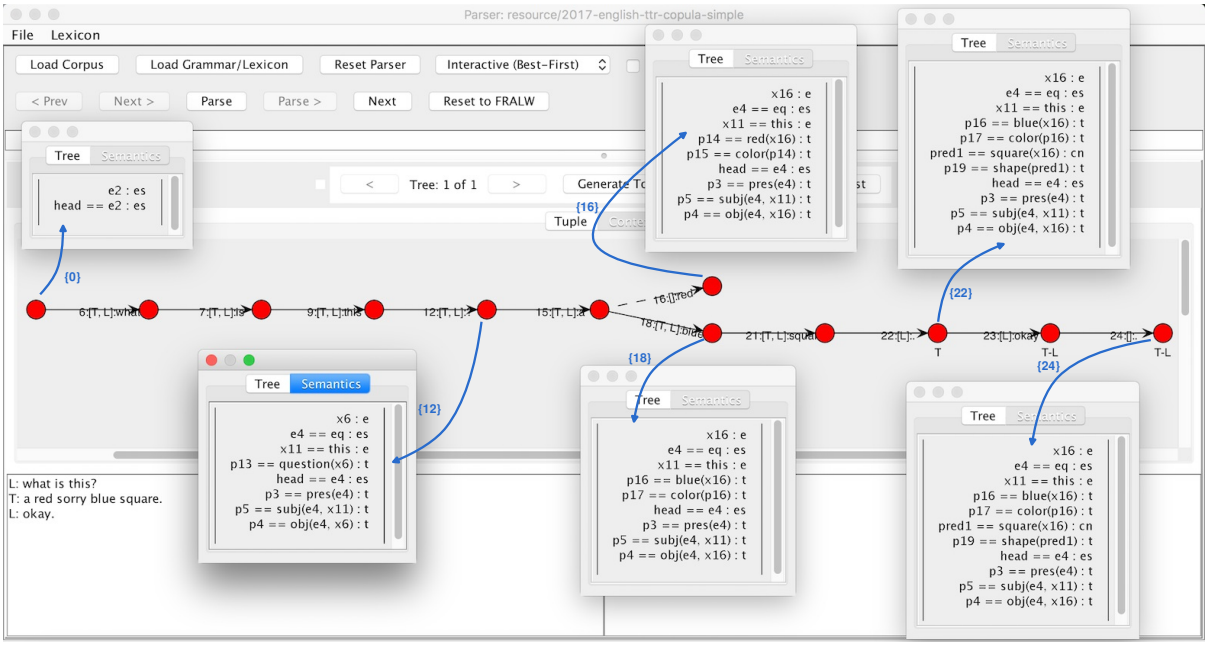


FIGURE 3.4: Incremental Processing with Dylan of a conversation between (T)utor and (L)earner "L: what is this? T: a red sorry blue square. L: okay."
1: when the tutor corrects himself this leads to backtracking on the DAG - the resulting Record Type (18) involves the manifest value of "blue" instead of "red" as in (16); 2: acceptances such as "okay" don't contribute any content, and just label the resulting node (24) as mutually agreed/accepted

3.2.1.2 Type Theory with Records (TTR)

Type Theory with Records (Cooper, 2005, Cooper and Ginzburg, 2015) is an extension of standard type theory. It has shown to be very useful in NL Semantics, as well as for dialogue modelling - see Cooper (2012), Ginzburg (2012), Purver et al. (2010) among others.

TTR is especially suited to the grounding problem, since it provides a natural bridging representation between perception and NL semantics, where meanings are definable as tightly linked to context, with both content and (perceptual) context expressible within the same type theoretical formalism. For example, Dobnik et al. (2012a), Larsson (2013) use TTR to model the semantics of spatial language, with spatial predicates grounded in visual classifiers. This is essential the same thing we do here in this thesis - but here in addition, TTR is integrated within an implemented, multi-modal dialogue system.

Record Types The logical forms in the TTR are specified as *record types* (RTs), which are constituted by a sequence of *fields*, in the form of $[l : T]$, in which l represents a particular, unique label and T represents its corresponding type (Cooper, 2005).

Records RTs can be witnessed (i.e. judged true) by *records* of that type, where a record is a sequence of label-value pairs $[l = v]$. We say that $[l = v]$ is of type $[l : T]$ just in case v is of type T.

$$R_1 : \begin{bmatrix} l_1 & : & T_1 \\ l_{2=a} & : & T_2 \\ l_{3=p(l_2)} & : & T_3 \end{bmatrix} \quad R_2 : \begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_{2'} \end{bmatrix} \quad R_3 : []$$

FIGURE 3.5: Example TTR record types

Fields can be *manifest*, i.e. given a singleton type e.g. $[l : T_a]$ where T_a is the type of which only a is a member; here, we write this using the syntactic sugar $[l_{=a} : T]$. Fields can also be *dependent* on fields preceding them (i.e. higher) in the record type (see Fig. 3.5).

Subtyping The standard subtype relation $\sqsubseteq$ (see more in (Fernández, 2006)) check can be defined for record types: $R_1 \sqsubseteq R_2$ if for all fields $[l : T_2]$ in R_2, R_1 contains $[l : T_1]$ where $T_1 \sqsubseteq T_2$. In Figure 3.5, $R_1 \sqsubseteq R_2$ if $T_2 \sqsubseteq T_{2'}$, and both R_1 and R_2 are subtypes of R_3. This sub-typing relation allows semantic information to be incrementally specified, i.e. record types can be indefinitely extended with more information/constraints. For us in this thesis, this is a key feature since it allows the system to encode *partial* knowledge about visual objects, and for this knowledge (e.g. object attributes) to be extended in a principled way, as and when this information becomes available.

Incremental Processing with DS-TTR Since TTR allows semantic content to be easily underspecified, and later extended through *subtyping*, it allows partial, sub-propositional, semantic content to be compiled for partial DS trees; which means that the *maximal semantic content* of an unfolding utterance is available after parsing every word - see Fig. 3.4. It also allows the context of the conversation (for example a Dialogue Game Board as in Ginzburg (2012)) to be specified with structured components.

Below, we use TTR to specify the non-linguistic, visual context of the conversation, thus allowing seamless integration between dialogue processing on the one hand, and visual processing on the other.

3.2.1.3 Integrating Vision and Language

Back to the original framework in Fig. 3.1 that shows how the various parts of the system interact: at any point in time, the vision system has access to an ontology of (object) types and attributes encoded as a set of TTR Record Types, whose individual atomic symbols, such as 'red' or 'square' are grounded in the set of classifiers within the vision system. The DS-TTR parser incrementally produces Record Types (RT), representing the

meaning jointly established by the tutor and the system so far. In this domain, this is ultimately one or more type judgements, i.e. that some scene/image/object is judged to be of a particular type, e.g. in Fig. 3.6 that the individuated object, $o1$ is a red square. These jointly negotiated type judgements then go on to provide training instances for the classifiers. In general, the training instances are of the form, $\langle O, T \rangle$, where O is an image/scene segment (an object), and T, a record type. T is then converted automatically to an input format suitable for specific classifiers (see example in Fig. 3.6 that provides the visual instance $\langle o1, \{red, square\} \rangle$ to visual classifiers).

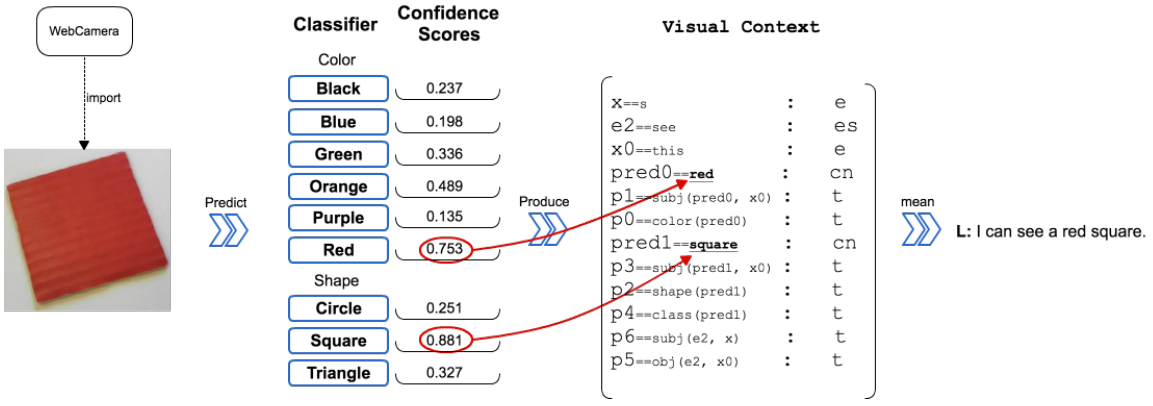


FIGURE 3.6: Example of how NL Semantics is grounded in Vision (where the system captures and classifies an object $o1$ with a set of attribute labels with certain confidence scores. The attribute classifiers for colour (e.g. *"red"*) and shape (e.g. *"square"*) ground the corresponding semantic atoms in the TTR visual context, used elsewhere for e.g. generation of descriptions, QA, or reference resolution)

The integrated system grounds the simple types (atoms) (e.g. "red", "square") and composes their output to construct the more complex type of the (visual) situation. This representation then acts as (1) the non-linguistic/visual context of the dialogue for DS-TTR, for definite reference/pronoun/indexical/ellipsis resolution; and (2) the logical database from which answers to questions about the objects' attributes are retrieved Yu et al. (2016a) - the question is parsed and its representation acts directly as a query on the non-linguistic/visual context to retrieve its answer (see Fig. 3.7 for a simple example where there is a circle and a square in the scene). Conversely, the system can form questions about the scene, where the teacher's answer then acts as a training instance for the classifiers (basic, atomic types) involved.

What sets our approach apart from previous work (as discussed in Chapter 2) is: **(1)** that we use a domain-general, incremental semantic grammar with principled mechanisms for parsing and generation; **(2)** Given DS model of dialogue (Eshghi et al., 2015), representations are constructed jointly and interactively by the tutor and system over the course of several turns, **(3)** visual perception and NL-semantics are modelled in a single logical formalism

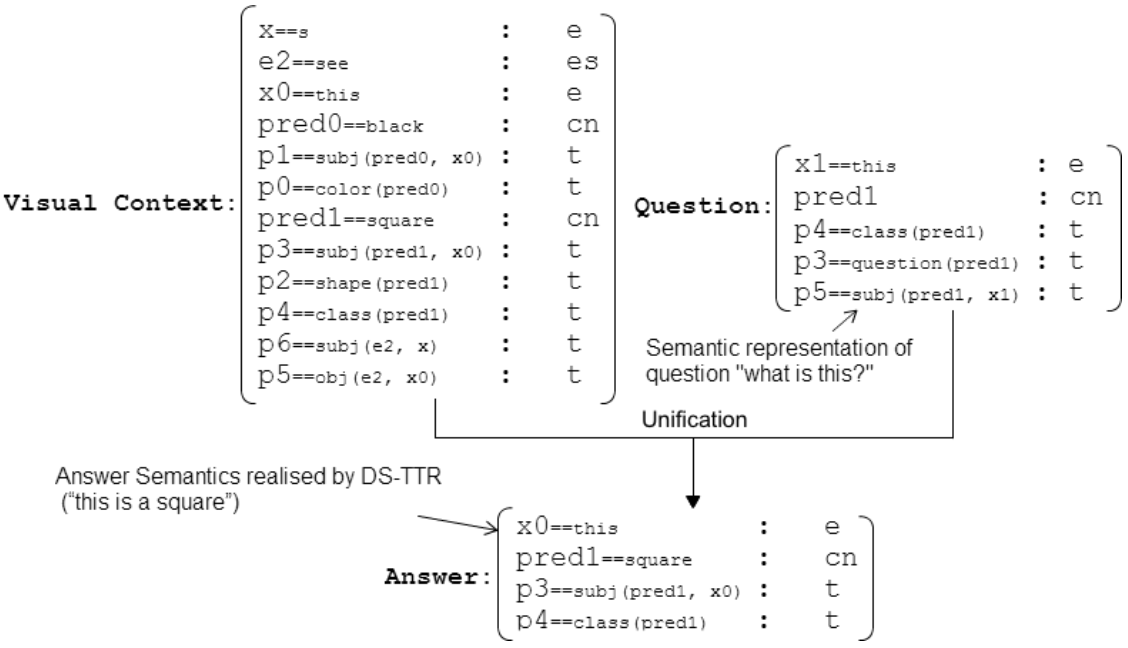


FIGURE 3.7: Answer retrieval from context

(TTR); **(4)** we effectively induce an ontology of atomic types in TTR, which can be combined in arbitrarily complex ways for generation of complex descriptions.

3.2.2 Dialogue Act Classification: Simple Spoken Language Understanding (SimpleSLU)

For NL understanding in dialogue, instead of deep semantic parsing using e.g. Dylan as above, the dialogue module can alternatively map utterances to Dialogue Acts (DAt) (Stolcke et al., 2000): abstract meaning representations which specify what action the utterance is performing, as well as parameters of that action. There is a large literature on the nature of Dialogue Acts, how they update Information States (see e.g. Larsson (2002), Traum and Larsson (2003)), their generality, standards (Bunt, 2006), etc. We do not go into any detail here, but only describe how we use DAts in an alternative, non-incremental version of the Dialogue Module.

Here, we introduce a simple Dialogue Act Tagging model, called Simple Spoken Language Understanding (SimpleSLU), that identifies and produces DAt representations of certain utterances using a set of hand-crafted principles. The SimpleSLU model performs a turn-level parse of the users utterance, without considering the previous dialogue context. It obtains the users intended action and related parameters using a pattern-matching algorithm that searches key patterns (e.g. words and phrases) in the utterance, and then automatically translates those patterns into DAt representations following the pre-defined rules. Given the visual-attribute learning task, we present a list of patterns consisting of dialogue acts,

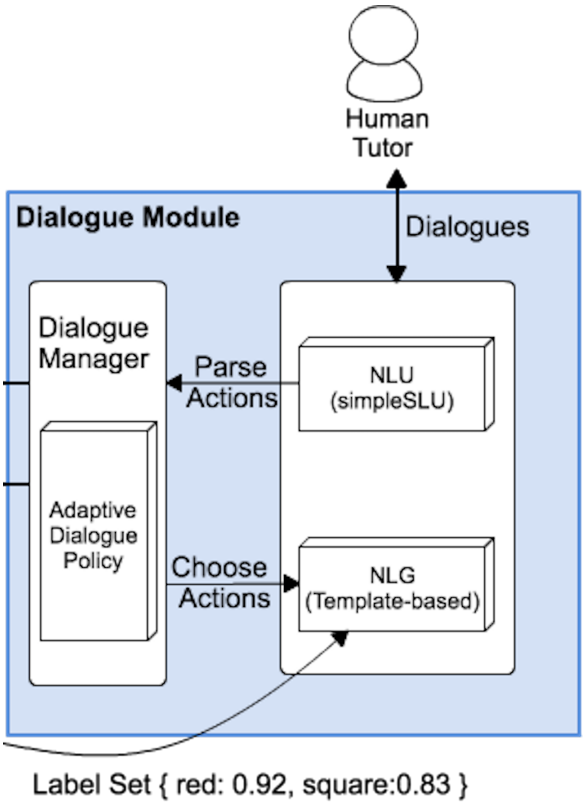


FIGURE 3.8: Architecture of Dialogue Module incorporating with the SimpleSLU

ontologies and specific attribute concepts (entities) (see examples in Table 3.1). For producing the final DAt representation for the single utterance, the model will finally package the sequence of translated tags into a full DAt representation (see the decision process in Fig. 3.9).

The model produces a single dialogue act representation by packaging these key patterns (see example in Fig. 3.9).

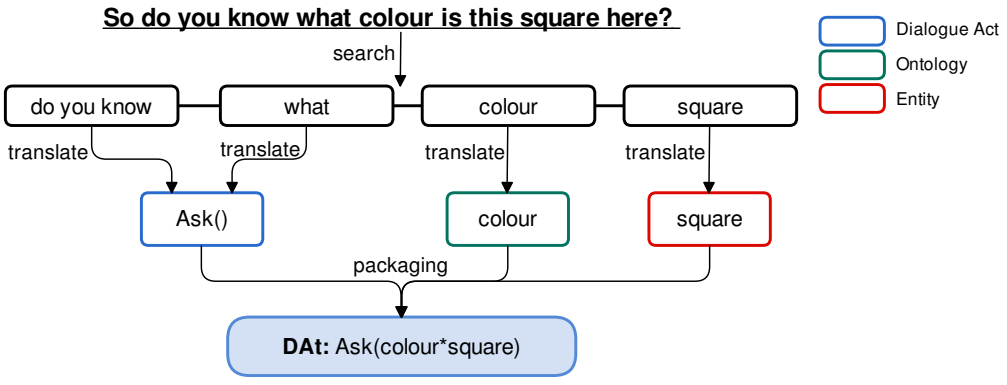


FIGURE 3.9: DAt decision process for "So do you know what colour is this square here?" with SimpleSLU

Category	Tag	Key Patterns
Dialogue Act	Ask	what, do you know, can you tell, what do you see, ...
	Reject	no, nope, incorrect, wrong, ...
	Polar	is this a, is this, is it, is it a, ...
	Accept	yes, yeah, yep, correct, right, good, good job, great job, well done, ...
	Repeat-Request	say again, again, repeat, what did you say, ...
	DoNotKnow	have no idea, don't know, ...
	...	...
Ontology	colour	colour, colours, color, colors
	shape	shape, shapes
Attribute Entity	colour	red, green, purple, black, yellow, blue, ...
	shape	triangle, circle, square, ...

TABLE 3.1: Examples of the Pattern List for Dialogue Act Identification with SimpleSLU (this list gives some examples of the DAt searching patterns. The dialogue-act tags come from annotations of human-human dialogues in the BURCHAK corpus (Chapter 5))

In human dialogue, a single dialogue turn may perform multiple actions, for instance, "please switch off the light and also close the window, when you leaving." performs the two actions: "*switch-off(light)*" and "*close(window)*". We have therefore designed the SimpleSLU model to be able to handle such multi-action turns in the DAt decision process, by looking at the nearest, available ontology/entity patterns in the pattern sequence (see more details in Chapter 8).

Symbol Grounding in SimpleSLU In contrast semantic parsing using the DS-TTR formalism, in SimpleSLU, the visual classifiers in the vision module ground visual attribute *words*, such as 'red', 'circle', etc., that appear here instead as parameters of the Dialogue Act representations (Stolcke et al., 2000) (e.g. inform(colour=red), ask(shape)) used in the framework.

This thesis deploys and tests the proposed learning agent with the two different NLU models presented above, and compare their performance on learning novel visual knowledge via NL conversation with human tutors in real time. We will also discuss the benefits, and limitations of these two very different approaches to language processing in the context of the symbol grounding problem (see more details in Chapters 8 and 9)

3.2.3 Semantic Parsing versus Dialogue Act Tagging

SimpleSLU is a fully hand-crafted model that maps an utterance into a dialogue intent, based on keyword pattern matching, following a list of pre-defined rules. This means that the SimpleSLU model can easily detect the user intent without deeply analysing the semantic and syntactic structures of the utterance. These rules can always be modified or extended to capture more complex conversational scenarios or cover more variations on the utterance

level. However, this method is essentially ad-hoc and fails to generalise to new data-sets or domains. The same thing can also be achieved using dialogue act classification techniques (Louwerse and Crossley, 2006, Webb, 2010), but these methods also suffer from similar drawbacks.

On the other hand, **Dynamic Syntax** is, by design, a domain-general semantic parsing framework that is unique in that it is word-by-word incremental, and provides general mechanisms for tracking the shared context of a conversation (see Section 3.2.1, but also Eshghi et al. (2015), Howes and Eshghi (2017)). Moreover, Type Theory with Records (TTR) has been argued to provide a seamless interface between perceptual semantics and NL semantics (Dobnik et al., 2012b, Larsson, 2015) and see above. The hybrid thus has the potential to provide a general framework for: (1) the development and learning of dialogue systems in general where NLU and NLG modules are transferable from one domain to another, and dialogue mangers can be learned from small amounts of unannotated dialogue data (see Eshghi et al. (2017), Kalatzis et al. (2016a)); and, (2) situated dialogue systems with grounded semantics that, we argue in the conclusion chapter, satisfy the requirements on such systems outlined in Chapter 2 (see Section 2.5, and Table 2.2).

Nevertheless, we must note here that developing new DS-TTR grammars requires linguistic expertise, and is potentially very time-consuming; however, we know from previous research that DS-TTR grammars can be automatically learned from data Eshghi et al. (2013), which can get around this problem. Another important practical issue in the application of grammar-based approaches in dialogue systems is access to wide-coverage grammars for dialogue so that there is little need for grammar development for each new domain. Since Eshghi et al. (2013) learn from a corpus of child-directed speech, the DS-TTR lexicon learnt is limited. More work is needed within the grammar induction task for learning from larger, more diverse corpora, leading to more wide-coverage grammars. In this thesis, we have had to extend existing DS-TTR lexicons manually, which has demanded substantial effort - but was acceptable given the clear advantages of such a method as outline above.

3.2.4 Dialogue Management

As discussed in Chapter 2, dialogue management (DM) is the component in charge of deciding what to do/say next in a dialogue system. The DM acts as a bridge that links to both NLU and NLG components: it accepts user's intents which are formulated in semantic representations (for instance, TTR record types or DAt representations) in NLU, and then outputs appropriate system responses or actions to NLG. The DM is also in charge of controlling the dialogue process and flow, e.g. producing appropriately timed backchannels, and turn-taking and management.

The Data-Driven Approach for Dialogue Management In recent decades, an increasing number of methods for addressing the DM problem have been. Among these DM approaches, data-driven approaches have become more popular and practical for dialogue modelling. Although some argue that data-driven approaches usually require a lot of time and effort on data annotation, the training process is completed automatic with little human supervision. One of the most prominent data-driven approaches is Reinforcement Learning (RL), based around Markov Decision Processes (MDPs) (Levin et al., 2000) or Partially Observable MDPs (POMDPs) (Williams and Young, 2007). The RL approach provides a general optimisation method via trial and error: they update their dialogue strategies by optimising some cumulative reward or cost functions given the current dialogue state after executing particular a particular action using some RL algorithms (for instance, Q-learning and SARSA).

Taking the SARSA algorithm as an example, a Markov decision process (MDP) is a tuple, $\{S, A, P, R, \gamma\}$, where:

- S: a finite set of states ($s_1, s_2, \ldots, s_t \in$ S);
- A: a finite set of actions ($a_t \in A$);
- T: transition dynamics $T(s_{t+1}|s_t, a_t)$, which map a state-action pair at time t to a distribution of states at time $t+1$;
- R: a local/global reward function given after transitioning from state s_t to state s_{t+1} by executing the particular state-action pair at time t;
- γ: a discount factor, scaled between 0 and 1, represents the difference in importance between immediate (local) rewards;

The SARSA algorithm - an on-policy algorithm - updates Q-values by interacting with the environment based on actions taken and the rewards received; where a Q-value represents the possible reward r_t gained in the next time step $t+1$, for taking particular an action a_t in state S, plus the discounted future reward gained from the next state-action observation (s_{t+1}, a_{t+1}), see Eq. 3.1.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a[r_t, \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{3.1}$$

The policy π is generally considered as a mapping from states to a probability distribution over actions: $\pi : S \rightarrow p(A|S)$. If the MDP is episodic, the state will be reset after each episode of length T, where sequences of states, actions and rewards constitute rollouts of

the policy, each of which contains cumulative rewards from the learning environment. (see Levin et al. (2000) for more details).

POMDP-based DMs on the other hand have had good success in handling uncertainties arising e.g. from speech recognition, Dialogue Act Classification, or semantic parsing (Gasic et al., 2013, Gasic and Young, 2014). We will not go into any more detail on POMDPs in this thesis, as it hasn't been employed.

Our implementation In this thesis, given the visual-attribute learning task, we build a DM that controls natural, spontaneous conversation in two ways: (1) hand-crafted policies to test specific hypotheses about the policies themselves; and (2) using Reinforcement Learning where the policy is learned from data. The former is implemented with a set of pre-defined dialogue rules and templates, which is applied to explore the possible and appropriate dialogue strategies by investigating their effects on the overall learning performance (Chapter 7). In (2), the policy is trained in interaction with a Simulated User, itself trained from real human-human conversations (see the next section & Chapter 5). This applies a tabular RL SARSA algorithm[2] to build a multi-objective MDP model to learn *when* and *how* to learn novel knowledge from humans. The "*when-to-learn*" policy is optimised to perform a form of *active learning* that allows the agent to ask for further information (e.g. WH/polar-questions and clarification requests) from humans through dialogue, *only when it needs to*. The "*how-to-learn*" strategy is a pure dialogue strategy that learns to conduct natural conversations with the tutor (more detail in Chapter 8).

The policy optimisation in the latter case is mainly applied to optimise the dialogue cost, reducing the human involvement within learning conversations. In our dialogue collection experiments (see Chapter 5), human participants, especially the learners, mainly focus their attention on learning/memorising unseen visual-attribute words, rather than minimising the time/conversational cost while learning new knowledge. But different to these learners, the agent in the project is required to avoid copying such human behaviours. Given a learning task, it should be able to take into account both factors: the learning/memorising accuracy and the cost, achieving good accuracy but with minimal dialogue cost.

3.3 Simulated Learning Environment

In order to build and evaluate the proposed interactive multi-modal architecture, this section introduces a fundamental and important part of the framework – a simulated learning environment (see Fig. 3.10) for the interactive learning task, in which the agent is required

[2]In this thesis, our state and action spaces were low-dimensional, so that we were able to use the tabular SARSA algorithm, rather than more powerful and complex approaches such as DRL, or POMDPs.

to learn novel visual concepts through conversation with a tutor simulation. This simulator must be able to resemble real human behaviours in NL conversation on the learning task. The learner (agent) and the simulated tutor will talk about both colour and shape of each single object chosen from a collection of visual objects in the learning process. The learner is able to describe or ask for useful information about particular visual attributes, and update its knowledge (i.e. classifiers) via the verbal feedback that it receives from the tutor.

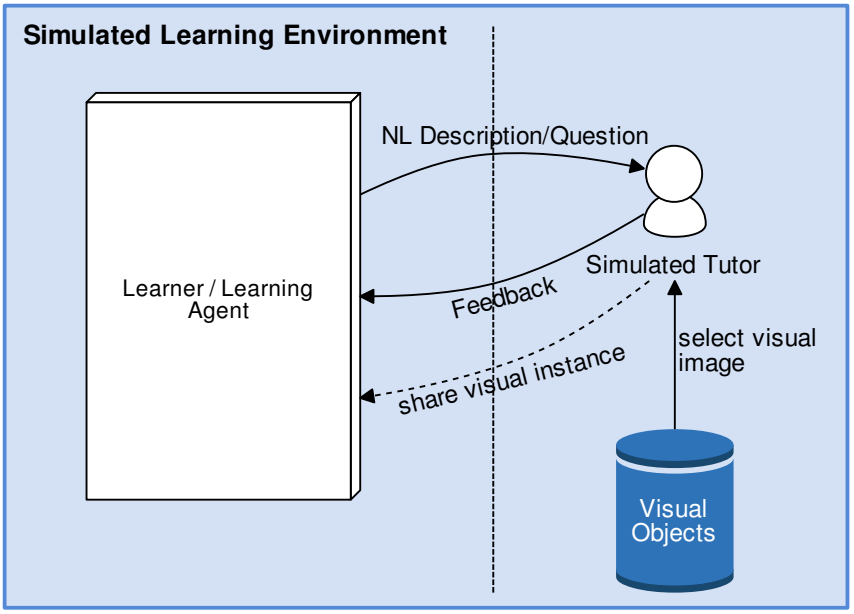


FIGURE 3.10: Architecture of the Simulated Learning Environment

3.3.1 Tutor Simulation

In this simulated learning environment, a tutor simulation plays an essential role in interacting with the learner/agent and teaching it how to recognise and describe certain visual attributes. The tutor will randomly select one visual object from the collection (see below) and then talk with the learner/agent using its visual attribute (colour and shape) in each single dialogue. In order to achieve a robust and practical learning agent, the tutor simulation here should be, not only responsible for simply teaching the learner novel visual concepts, but also for training and evaluating the learner on its capability for processing natural, human-like conversations.

In this thesis, we introduce a novel and generic n-gram framework in support of building a user simulation for training and testing dialogue systems. The model is learns to generate coherent user responses given particular conditions (system moves and additional goal-driven conditions) utterance-by-utterance, action-by-action and even word-by-word. For lowering

the mismatch risk, the framework deploys a back-off method that allows the model to backtrack to smaller n-grams when it cannot find any n-grams matched to the current word sequence and conditions. It also employs a nearest-neighbour algorithm for searching the n-gram matches for the unseen system moves by calculating the Hamming distance between each pairs of n-grams (see details in Chapter 6). The model in this thesis is trained either from synthetic dialogue examples or from a corpus of real human conversations in our concept learning domain (the BURCHAK corpus, see Chapter 5).

Since natural, spontaneous dialogue (like in the BURCHAK corpus) is inherently incremental (Crocker et al., 2000, Ferreira, 1996, Purver et al., 2009a), it usually contains a variety of characteristic, incremental dialogue phenomena such as self-repair and repetition, hesitations, fillers etc., which are likely to be interactionally and semantically consequential for the dialogue: they affect how the conversation partner (the dialogue agent) adapts to users and coordinates their moves over time. Although there has recently been a surge of previous work showing interests and progress on the implementation of user simulations, they seldom pay any attention to these incremental dialogue phenomena in more natural, human-like conversations. Hence, in order to solve the issues for the learning/dialogue agent, one of the essential challenges and also milestone of the proposed user model is that it can simulate natural, spontaneous conversations by *inserting* or *predicting* incremental phenomena (e.g. self-repairs, repetitions, pauses, and fillers) from the BURCHAK corpus (See Chapters 5 & 6)

3.4 Chapter Summary

We presented an **Interactive Multi-modal Framework**, in support of building teachable robots/interfaces that are able to grounding symbols in Natural Language into aspects of the physical environment and vice versa through NL interaction with real humans. The framework takes into account all properties discussed in the previous chapter: the agent *compositionally* learns *visual-attribute* concepts through *Natural Language conversations* with real humans following a optimised strategy, *incrementally*, over time. This framework is modular, which allows the different components to be plugged in and out without affecting the overall integrity of the framework. For instance, the framework is be able to be integrated with, either a simple dialogue act tagging model (SimpleSLU) or the DyLan semantic parser. It is also able to deploy either hand-crafted rule-based dialogue managers or optimised strategies using Reinforcement Learning. In the rest of this thesis, we will discuss and explore the most appropriate approach for each module/component in the framework.

We also presented the design of the simulated learning environment in which the agent learns to identify and describe visual objects using their attributes (e.g. colour and shape) through interaction with the simulated tutor, where the simulation is built using an n-gram approach (Chapter 6), either from a set of synthetic dialogue examples, or based on realistic conversations (Chapter 5).

Chapter 4

Comparison of Classification Models for Learning Visual Attributes

This chapter explores appropriate data-driven approaches for learning/identifying low-level properties (e.g. colour and shape), which are applied to distinguish individual perceptual objects in the physical world. In order to learn those novel visual attributes through dialogue with humans in real time (as concerned in this thesis), it requires a learning system/interface that can work in an update-incremental fashion, without re-computing previous increments, and can learn novel visual knowledge efficiently with fewer examples.

In the following sections, we briefly review a set of existing classification models for learning visual attributes of real-world objects (in section 4.1), where those approaches are considered from two main dimensions in the physical world: 1) single- or multi-label learning task, and 2) offline or online learning task. That is followed by two experiments (see section 4.2 and section 4.3) for investigating 1) how the task of visual-attribute learning/classification in an incremental learning process can be addressed and 2) what classifier models can fit into such learning task.

Finally, section 4.4 summarizes explorations of visual-attribute classification approaches in this chapter. Since the task of interactively learning visual scene in real time is concerned about a trade-off between good learning performance (recognition accuracy) and learning efforts (time consumption), a simple but efficient learning method (an optimised logistic regression SVM with the Stochastic Gradient Descent algorithm (called SGD-SVM as below)) is integrated with the interactive framework in Chapter 3.

4.1 A Review of Visual-attributes Classification Approaches

Previous Work	Classification Model	Single-/Multi-label	Type of Training data
(Zhang, 2004)	SGD-SVM	Single-label	Dynamic (online)
Zhang and Zhou (2007)	ML-kNN	Multi-label	Statistic (offline)
(Farhadi et al., 2009)	Linear SVM	Single-label	Statistic (offline)
(Sirinart Tangruamsub and Hasegawa, 2011)	SGD-SOINN-SVM	Single-label	Dynamic (online)
(Sun et al., 2013)	K-SVD	Single-label	Statistic (offline)
(Silberer et al., 2013)	L2-loss Linear SVM	Single-label	Statistic (offline)
(Kong et al., 2013)	TRAM	Multi-label	Statistic (offline)
(Kristan and Leonardis, 2014)	oKED	Single-label	Dynamic (online)
(Thomason et al., 2016b)	Quadratic-kernel SVM	Single-label	Statistic (offline)
(Kennington, 2016)	Word-as-Classifer	Single-label	Dynamic (online)

TABLE 4.1: Review of previous work on the visual attribute classification task

This section describes the previous work for learning to recognise real-world visual objects and their relevant properties, for example colour, shape, texture and material. We briefly review their classification approaches from two dimensions (see Table 4.1), 1) learning with single or multiple labels and 2) learning with or without relearning previous examples, as detailed below:

4.1.1 Single-label versus Multi-label Classification

In the past decades, there has been significant interests and progresses on learning to describe visual objects/attributes from the external word. Generally, there are two main strands of work that address this problem: 1) learns one classifier for each single attribute/label and 2) learns a group of labels (multiple) for each single object. Among the classifier modes in the former stand of work, Support vector machine (SVM) is a simple but powerful approach that is commonly to identify functional and non-functional features by integrating with variety of optimisation algorithms, for instance, logistic regression (Farhadi et al., 2009, Silberer and Lapata, 2014), kernel method (Kennington, 2016, Lampert et al., 2014, Thomason et al., 2016b), and stochastic gradient descent (Bottou, 2012, Zhang, 2004). Apart from the SVM model, there are also some outstanding methods for addressing the single-label classification problem.

For example, Abdelsamea et al. (2015) introduced a new classifier structure based on original Self-Organized Map (SOM) – improved SOM model (*i*SOM) – that focuses on supervised learning tasks in the field of object-feature recognition and contributes to the achievement of high classification accuracy during a short-time period. Specifically, the *i*SOM model mainly contains two parts: a new node structure and weight updating. In the node structure, each node is represented with a set of connection weights ($w = \{w_0, w_1, \cdots, w_n\}$, where n represents the number of attributes), and a set of wining class counters (WCC_m $c_m = \{c_1, c_2, \cdots, c_m\}$, where m for the number of classes.) (Abdelsamea et al., 2015). In the

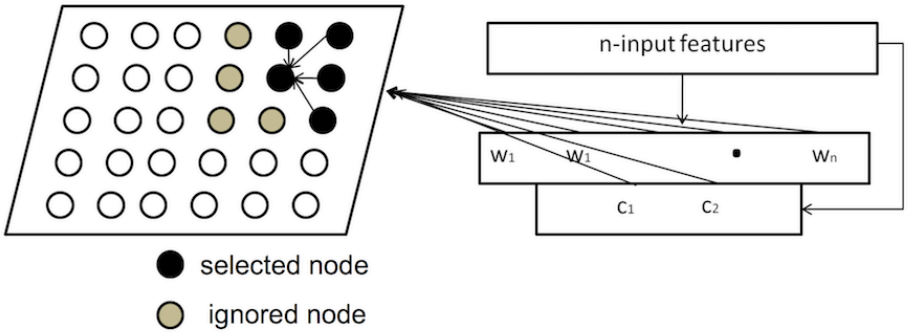


FIGURE 4.1: "The architecture of *i*SOM: the new node structure and weight updating" (Abdelsamea et al., 2015)

learning process, the vector WCC performs as a voting criteria, i.e. moving the nodes with maximum WCC_i to the best-matching unit (BMU), and leaving nodes exactly from other classes. After finding the BMU, the model can increase the WCC_i by 1 for i-th class, which also increase the confidence "that the node is targeted by an example of class i" (Abdelsamea et al., 2015). This confidence will help to identify the winning node by calculating its similarity with the input example. Finally, the model selects nodes which are mostly targeted by examples from the same class, and then update their connection weights to bring them closer to the same cluster, and pull others left (see more details in (Abdelsamea et al., 2015)).

On the other hand, the second stand of work attempt to solve the issue of learning/classifying visual attributes of the real-world objects as a multi-label learning task, for instance Zhang and Zhou (2007) and Kong et al. (2013). This line of work usually design their models to predict a bag of attributes instead of particular attributes to alleviate the confusion from unclear labels.

Zhang and Zhou (2007) suggest a fully supervised multi-label learning model based on the k-Nearest Neighbour algorithm (ML-kNN), which supports a prediction of a label set for unknown instances. It has previously been used for scene labelling with 5 labels (sunset, desert, mountains, sea, trees) and reached a Precision of 0.8. (see more details about how this model is formulated in Section 4.2.1.1)

Kong et al. (2013) develop a semi-supervised model – Transductive Multi-label Learning model (TRAM) – to determine the label set of a novel instance based on utilised information from both labelled and unlabelled data. This model is designed for addressing two key issues during the object recognition process: 1) **Lack of labelled data**: as traditional multi-label classification models, as supervised methods, rely on a large amount of annotated data from the real-world, they usually suffer from a lack of training samples resulted from high labelling costs; and 2) **Multiple labels**: each instance can be associated with multiple

concepts, e.g. "red square" and "blue circle". In the transductive learning process, this model makes use of both labelled and unlabelled data to estimate the label set cardinality, and then predict the final label set with ranked labels using the estimated cardinality ((see more details in Section 4.2.1.2)). Fu et al. (2014) extend this TRAM model with a zero-shot learning algorithm (Lampert et al., 2014) – TraMP – that attempts to learn visual objects instead of attributes by exploiting multi-label corrections without/with few training data.

4.1.2 offline versus online Classification

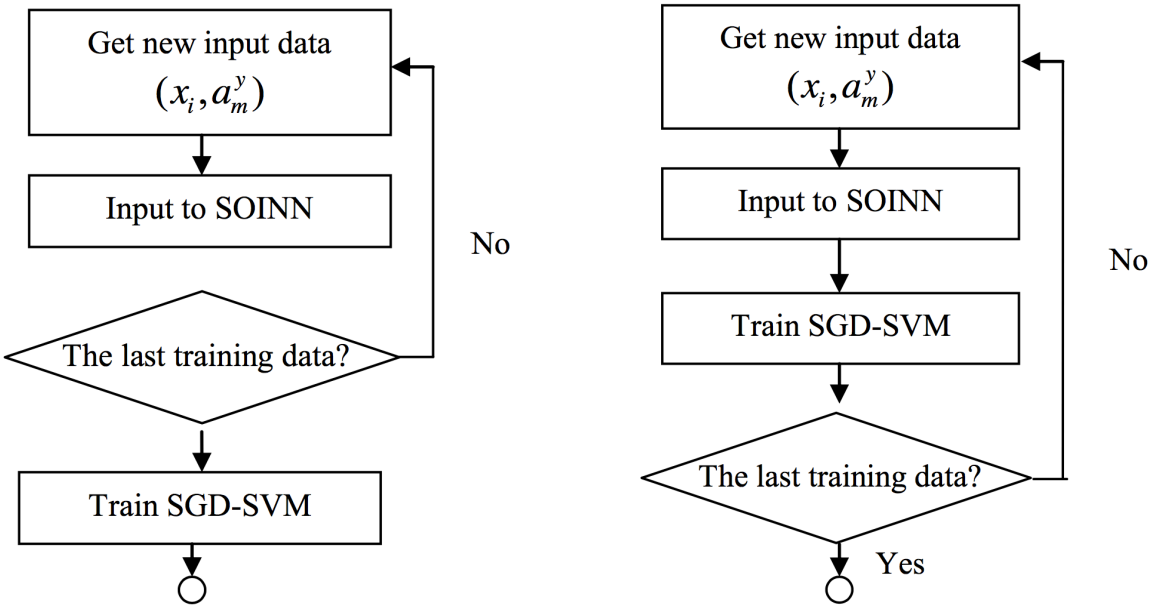
Another dimension along which work on attribute learning task can be compared is whether the visual classifiers are learned with statistic training examples (offline) as in (Farhadi et al., 2009, Lampert et al., 2014, Silberer et al., 2013, Thomason et al., 2016b) or dynamic data (online), for instance, (Bottou, 2012, Kennington, 2016, Kristan and Leonardis, 2014, Sirinart Tangruamsub and Hasegawa, 2011). Since our goal of this thesis is to learn visual attributes with humans in the real time, we here mainly present previous work that runs with the latter method.

(Zhang, 2004) implemented a simple but very efficient method that discriminatingly learns linear classifiers under convex loss functions and Stochastic Gradient Decent (SGD), called SGD-SVM. Different with previous approaches that needs to be retrained when new information is added, the SGD-SVM model can learn and classify new training example, without re-computing previous increments. This model applies a single binary classifier model that takes into account each visual class or attribute as an independent category.

Sirinart Tangruamsub and Hasegawa (2011) introduced a new fast online incremental classification model based on the self-organising incremental neural network (SOINN)[1] by Furao and Hasegawa (2006). They proposed two versions of SOINN-SVM classifier models, both of which are also trained using the SGD-SVM model above, but in different ways (see Fig. 4.2).

The main difference between two SOINN-based SVM models is whether the SGD-SVM model is trained instantaneously after a new sample (x_i, a_m^y) is fed (where "x_i is an input image feature of class y and a_m^y is the binary value of the m-th attribute"). As illustrated in the figure, the SGD-SOINN-SVM model incrementally trains both the cluster and SVM model while new input is added. Compared with that, the SOINN-SVM model will let the SOINN cluster grow incrementally until the last example is input into the model, and then

[1]Self-Organising Incremental Neural Network (SOINN) is an unsupervised classification model that supports inference of classes for non-stationary data and represents the topological structure of a probabilistic distribution over inputs. (see more details about how it works in Furao and Hasegawa (2006))



(A) SOINN-SVM (B) SGD-SOINN-SVM

FIGURE 4.2: Overview of two SOINN-based SVM classifier models

train the SGD-SVMs using the cluster nodes instead of image features. Sirinart Tangruamsub and Hasegawa (2011) set an experiment to compare both proposed models with the standard SGD-SVM: although the SGD-SVM learns and classifies new instances faster than the proposed models, both SOINN-SVM and SGD-SOINN-SVM models were performing better than the SGD-SVM mode.

Kristan and Leonardis (2014) designed a new supervised approach to classifier estimation – online discriminative Kernel Destiny Estimation (oKDE) – that continuously estimates and builds the probability destiny functions (*pdf*) from data by observing a single instance at a time. This model estimates *pdf* using a mixture of Gaussian and automatically adjusts model complexity on the target distribution. The oKDE model supports update from both positive samples (for learning) and negative samples (for unlearning). It can make model effectively distinguish different attributes in the same category/class. Figure 4.3 outlines how the model is applied in adapting a three-class classifier (see more details in (Kristan and Leonardis, 2014)).

However, different with standard binary classifiers, the oKDE requests a predefined classification category before learning specific labels, which means that system needs to know what labels can be grouped together for particular class, i.e. "*red*" and "*blue*" are in the same category of **Colour**, but "*square*" is for **Shape**.

Kennington (2016) proposed a simple discriminative classifier model that associates the single word (for instance, 'red', 'f-shaped' and etc.) with the visual classifiers, which is

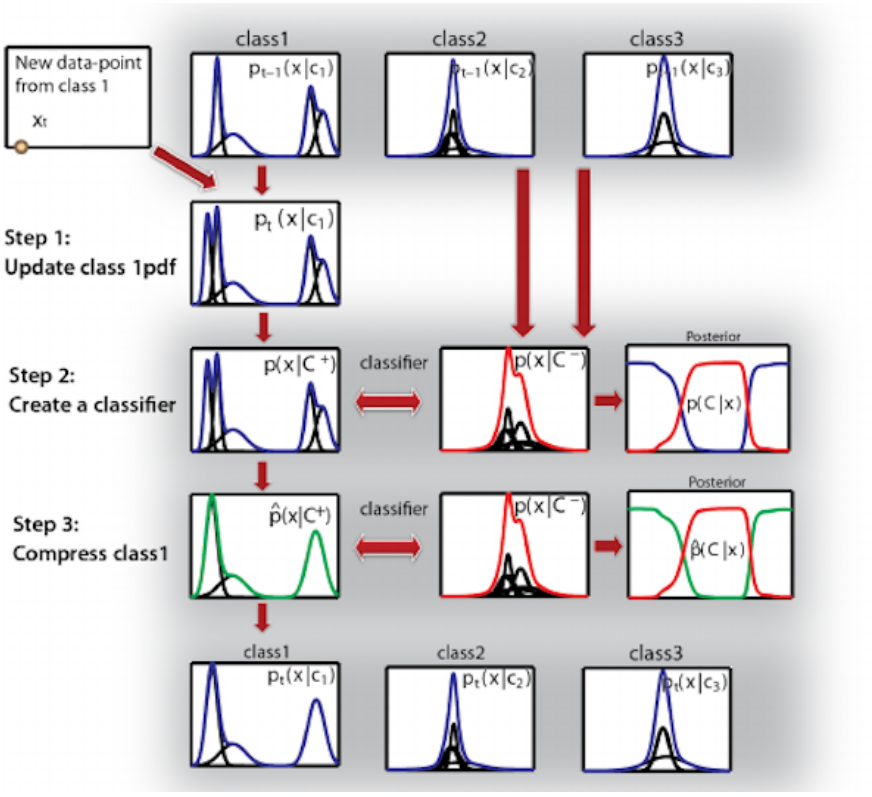


FIGURE 4.3: "Illustration of he main three steps in the oKDE model. The example shows a three-class model in which the first class is updated by a new observation and compressed. While the distributions change significantly, the classifiers posterior does not." (Kristan and Leonardis, 2014)

similar to what we do in the thesis. Instead of words, we train the classifiers to predict atomic items in formal semantics given low-level visual features. Here, Kennington (2016) trained a binary logistic regression classifier (SVM) for each word that estimates the probability p_w of how well a candidate object fits to the particular word given its representation via (simple) visual features (X) (see the visual representation in Figure 4.4), as below:

$$p_w(X) = \sigma(w^T X + b) \tag{4.1}$$

where w represents the weight vector that is learned and σ represents the logistic regression function. The classifier model is a one-layer neural network.

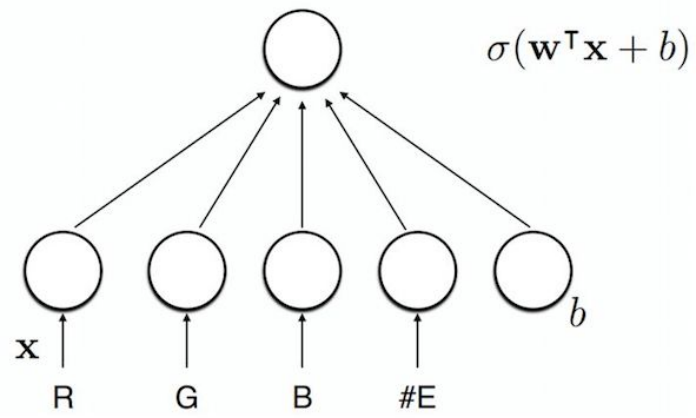


FIGURE 4.4: Representation as a 1-layer Neural Network (R,G and B presents colour features, #E is the number or edges) (Kennington, 2016)

4.1.3 Summary

In this section, we reviewed previous work[2] that learns to identify visual attributes of the real-world objects, using either single- or multi-label classification models. We also discussed several existing classification approaches that learns with dynamic training examples incrementally, over time. For some technical and commercial reasons, we unfortunately cannot replicate some approaches mentioned above. We finally selected four of them (including ML-kNN, TRAM, Linear SVM and SGD-SVM) that can be easily and properly re-implemented/deployed for further tests. Later in this chapter, we present two experiment to explore the most appropriate learning algorithm fitting into the interactive learning scenario as below.

4.2 Experiment 1: Learning Visual-Attribute with Multi-label Classification Models

Following the review of previous work, learning visual attribute of real-world objects can be viewed as a typical multi-label learning task. For instance, in the physical world, each perceptual object may consists of several predefined properties, such as colour, shape, material as well as functional features, such as "this is a red elongated pen", meanwhile, an object may contain more than one feature in an individual property category, e.g. "this mug has vertical stripes of blue and white" or "it is a black square with a red cross".

[2]Although there are more related work that learns visual classifiers using Deep Learning techniques, we do not discuss any of these techniques, because none of them is suitable for an interactive learning task – learning novel knowledge incrementally with no or small amount of data. Such deep learning approaches always requires a large number of training examples (i.e. millions of images) with high-level quality, which contradicts our research goals in this project.

The experiment presented in this section therefore aims at testing the performance of multi-label learning models on classifying real-world objects and their attributes with little knowledge. For simulating an incremental an incremental learning process with human tutors, in this experiment, we also evaluate the performance increase of two models as more training instances are fed to them.

4.2.1 Classification Approach

In this section, we will mainly describe two approaches from the previous work for addressing multi-label learning task.

- Multi-label $irstk$-Nearest Neighbours (ML-kNN), proposed by Zhang and Zhou (2007), is an extension of the standard k-Nearest Neighbour algorithm. ML-kNN predict a set of labels of each novel instance by identifying its k-nearest neighbours within the training set and then utilising the maximum a posteriori principle upon derived statistical information from the k neighbours' labels.
- Transductive Multi-label Learning (TRAM) by Kong et al. (2013), as an extension of transductive learning, fully takes advantage of both labelled and unlabelled data to address the multi-label annotation problem. It proposes a semi-supervised model to determine a label set of a novel instance based on utilised information from both labelled and unlabelled data.

More details are shown in the following sections:

4.2.1.1 ML-kNN

We firstly attempt to apply ML-kNN by Zhang and Zhou (2007) to learn multiple attributes for a single instance. ML-kNN considers k-nearest neighbours in the vector space, given an instance x as well as corresponding label set $Y \subseteq y$. $\vec{y_x}$ is defined as a class vector for instance x, where its $l-th$ value $\vec{y_x}(l)$ takes 1 if $l \in Y$ and 0 otherwise. Additionally, $N_{(x)}$ represents a set of k-nearest neighbours of x discovered in the training set. Therefore, a *Membership Counting Vector* can be defined based on label sets of neighbours (Zhang and Zhou, 2007), as:

$$\vec{C_x}(l) = \sum_{a \in N_{(x)}} \vec{y_x}(l), \; l \in y, \tag{4.2}$$

where $\vec{C_x}(l)$ represents how many neighbours of x that belongs to the $l-th$ class.

$[\vec{y}_t, \vec{r}_t]$=ML-KNN(T, K, t, s)

%Computing the prior probabilities $P(H_b^l)$

(1) **for** $l \in \mathcal{Y}$ **do**

(2) $P(H_1^l) = (s + \sum_{i=1}^{m} \vec{y}_{x_i}(l)) / (s \times 2 + m)$; $P(H_0^l) = 1 - P(H_1^l)$;

%Computing the posterior probabilities $P(E_j^l|H_b^l)$

(3) **Identify** $N(x_i)$, $i \in \{1, 2, \ldots, m\}$**;**

(4) **for** $l \in \mathcal{Y}$ **do**

(5) **for** $j \in \{0, 1, \ldots, \mathrm{K}\}$ **do**

(6) $c[j] = 0$; $c'[j] = 0$;

(7) **for** $i \in \{1, 2, \ldots, m\}$ **do**

(8) $\delta = \vec{C}_{x_i}(l) = \sum_{a \in N(x_i)} \vec{y}_a(l)$;

(9) **if** $(\vec{y}_{x_i}(l) == 1)$ **then** $c[\delta] = c[\delta] + 1$;

(10) **else** $c'[\delta] = c'[\delta] + 1$;

(11) **for** $j \in \{0, 1, \ldots, \mathrm{K}\}$ **do**

(12) $P(E_j^l|H_1^l) = (s + c[j])/(s \times (\mathrm{K} + 1) + \sum_{p=0}^{\mathbf{k}} c[p])$**;**

(13) $P(E_j^l|H_0^l) = (s + c'[j])/(s \times (\mathrm{K} + 1) + \sum_{p=0}^{\mathbf{k}} c'[p])$;

%Computing $\vec{y}_t$ and $\vec{r}_t$

(14) **Identify** $N(t)$**;**

(15) **for** $l \in \mathcal{Y}$ **do**

(16) $\vec{C}_t(l) = \sum_{a \in N(t)} \vec{y}_a(l)$;

(17) $\vec{y}_t(l) = \arg\max_{b \in \{0,1\}} P(H_b^l) P(E^l_{\vec{C}_t(l)}|H_b^l)$;

(18) $\vec{r}_t(l) = P(H_1^l|E^l_{\vec{C}_t(l)}) = (P(H_1^l)P(E^l_{\vec{C}_t(l)}|H_1^l))/P(E^l_{\vec{C}_t(l)})$

$= (P(H_1^l)P(E^l_{\vec{C}_t(l)}|H_1^l))/(\sum_{b \in \{0,1\}} P(H_b^l)P(E^l_{\vec{C}_t(l)}|H_b^l))$;

FIGURE 4.5: Pseudo code of ML-kNN (Zhang and Zhou, 2007)

Given a single test instance t, ML-kNN can identify its k-nearest neighbours $N_{(t)}$ within the training set. Let H_1^l define the scenario where instance t has a label l, whilst H_0^l defines the scenario that t does not have the label l. Moreover, let $E_j^l (j \in 0, 1, ..., k)$ represent the situation in which there are j instances that have a label l among the k-nearest neighbours of t. Thus, a category vector $\vec{y}_t$ can be determined based on the $\vec{C}_x(l)$ using maximum a posteriori principle, as shown below:

$$\vec{y}_t(l) = \mathsf{argmax}_{b \in 0,1} P(H_b^l|E^l_{\vec{C}_x(l)}),\ l \in y, \tag{4.3}$$

Using the Bayesian rule, equation 4.4 may be updated as:

$$\vec{y_t}(l) = \text{argmax}_{b\in 0,1} \frac{P(H_b^l)P(E^l_{\vec{C_x}(l)}|H_b^l)}{P(E^l_{\vec{C_x}(l)})} = \text{argmax}_{b\in 0,1} P(H_b^l)P(E^l_{\vec{C_x}(l)}|H_b^l), \tag{4.4}$$

where $\vec{y_t}(l)$ considers both the prior probabilities $P(H_b^l)(l \in y, b \in 0, 1)$, as well as the post probabilities $P(E^l_{\vec{C_x}(l)}|H_b^l)(j \in 0, 1, ..., k)$, which are estimated directly from the training set. There is a pseudo-description for ML-kNN shown in Fig 4.5

As shown in Fig 4.5, ML-kNN estimates the prior probabilities $P(H_b^l)$ on steps (1) and (2), and then estimates the post probabilities $P(E^l_{\vec{C_x}(l)}|H_b^l)$ from steps (3) to (13), where c[j] is applied in each iteration of l to count the number of training instances with the label l (Zhang and Zhou, 2007). Its k-nearest neighbours includes j instances with label l. Finally, steps from (14) to (18) output the final results based on the estimated probabilities using the Bayesian rule (Zhang and Zhou, 2007).

4.2.1.2 TRAM

Following the description of the TRAM model (Kong et al., 2013) in section 4.1, it concentrates on solving the problem of predicting a set of multiple labels for groups of unknown samples based on a limited number of known examples and a large amount of unseen data. The model learns to identify the label sets of the unlabelled instances simultaneously by "utilizing the information from both labelled and unlabelled data" (Kong et al., 2013).

Generally, (Kong et al., 2013) address such multi-label prediction task in two essential steps, as below:

1. **Estimation of Label Concept Composition**[3]: Since all unlabelled instance will be estimated at the same time, and similar instances may contain similar label concepts in their own label sets, this step aims at jointly estimating the composition of label concepts on each unlabelled instance.

   For address this step, Kong et al. (2013) denote the concept composition for each instance x_i as $\alpha_i = (\alpha_{i1}, \alpha_{i2}, ..., \alpha_{im})^T$, in which a_{ij} represents the proportion of a label concept l_i in an instance x_i. As there is only concept composition information available on training instances, Kong et al. (2013) defined the ground-truth concept composition $\bar{a_{ij}}$ for each labelled instance x_i as below:

[3]Here, Kong et al. (2013) define the label concept composition as a multi-label instance, i.e. an instance x_i with a label set Y_i containing a set of multiple label concepts. For example, if we have a box, with 20% of the box coloured the label concept "red" (l_1), 30% in the label concept "blue" (l_2) and 50% in "green" (l_3), we can tell that the instance x_i has a label set $\{l_1, l_2, l_3\}$, and its label concept composition is $\{l_1 : 0.2, l_2 : 0.3, l_3 : 0.5, \cdots, l_n : 0\}$

$$\bar{\alpha_{ij}} = \begin{cases} \frac{1}{|Y_i|}, & \text{if } l_i \in Y_i \\ 0, & \text{otherwise} \end{cases} \quad (i \in L), \tag{4.5}$$

where Y_i represents the label set for a multi-label instance x_i, and L represents a set of label concepts. All label concepts within the label set of instance x_i have equal weights for concept composition.

For optimising the estimation of the concept composition, Kong et al. (2013) apply k-nearest neighbour (kNN) to optimally characterize the correlations between similar instances in their feature space. Through the kNN search, a sparse $n \times n$ matrix W is defined to indicate the similarity among neighbouring instances, as below:

$$\bar{W_{iz}} = \begin{cases} \frac{1}{Z_i} \exp(-\frac{\|x_i - x_z\|^2}{2\sigma^2}), & \text{if } z \in N_i \\ 0, & \text{otherwise} \end{cases}, \tag{4.6}$$

where $N - i$ indicates the index set of i-th instance's k-nearest neighbours, $\|\cdot\|$ represents the Euclidean distance, and σ refers to the average distance estimated between instances. $Z_i = \sum_{z \in N_i} \exp(-\frac{\|x_i - x_z\|^2}{2\sigma^2})$ is a normalisation term for guaranteeing that $\sum_z W_{iz} = 1$ for instances.

On the other hand, Kong et al. (2013) also design a closed-form solution for calculating alpha values α_{ij} from , which are used to predict a set of labels for each unlabelled instance in the next step. Here, they partition matrix A (i.e. $A = I - W$, where matrix I represents features of unlabelled instances, matrix W represents similarities among neighbouring instances) and and $a_{(j)}$ vectors into blocks based on labelled and unlabelled data: $A = \begin{bmatrix} A_{LL} & A_{Lu} \\ A_{uL} & wA_u \end{bmatrix}$ and $a_{(j)} = \begin{bmatrix} a_{L_j} \\ a_{u_j} \end{bmatrix}$, $(j = 1, ..., m)$, where L represents labelled data, and u represents unlabelled data.

Following a series of algorithm (see details how the closed-form solution calculates in (Kong et al., 2013)), an optimal solution of the alpha values for unlabelled instances can be computed using a linear equation below:

$$A_{uu} a_{uj} = -A_{uL} \bar{a}_{Lj}, \tag{4.7}$$

where, Kong et al. (2013) pointed out that the $\bar{a}_{uj}$ can be guaranteed to exist and to be unique with values ranged between 0 and 1.

2. **Label Set Prediction**: Given the estimated label concept composition and a *limited number of known examples*, types of the estimated label sets sometimes are less likely to have representative data in the training set. Here, it mainly focuses on address the issue of how to predict new label sets under such condition.

$(Y_U, \boldsymbol{\alpha}_\mathcal{U})$ = TRAM(X, $Y_\mathcal{L}$)

Input:

X : $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ encoding features of the whole data set

$Y_\mathcal{L}$: $(Y_1, \cdots, Y_l)$ encoding labels of training set

Process:

1 Construct kNN graph among instances.

2 Initialize the similarities on each edge as $W_{iz} = \exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_z||^2}{2\sigma^2})$ and normalize to $\sum_z W_{iz} = 1$;

3 Determine the $\boldsymbol{\alpha}_\mathcal{U}^j$ values for all unlabeled data by solving the linear system in Eq.6;

\# Transductive version:

4 Compute sorted label list on each unlabeled instance using optimal alpha values in Step 3;

5 Determine the optimal number of labels on each instance by solving the linear equation in Eq. 10.

Output:

$Y_\mathcal{U}$: the predicted labels for unlabeled instances.

$\boldsymbol{\alpha}_\mathcal{U}$: the alpha value outputs for unlabeled instances.

FIGURE 4.6: Pseudo code of TRAM (Kong et al., 2013)

In this step, Kong et al. (2013) define a transductive label set prediction method that can utilise information from both labelled and unlabelled instances using the optimal alpha value for an unlabelled instance x_i described above. A list of potential labels for x_i will be sorted based on their alpha values in descending order, i.e. the larger the alpha value is, the more likely x_i can be assigned the corresponding label. Kong et al. (2013) denote θ_i as the number of labels in the label set for instance x_i, because values of θ_i, (non-negative integers) are determined based on the grounded truth of their label sets, i,e, $\theta_i = |Y_i|, (i \in L)$.

Similar to the optimisation issue of composition estimation above, an optimal solution can be fond to sign the θ_i values with a linear equation as below:

$$A_{uu}\theta_u = -A_{uL}\theta_L, \tag{4.8}$$

Kong et al. (2013), where $\theta = (\theta_1, ..., \theta_n)^T = \begin{bmatrix} \theta_L \\ \theta_u \end{bmatrix}$. The optimal solution (θ_i^*) can be applied for predicting label set of unknown instance x_i.

Note that, as the TRAM model contains massive and complex mathematical algorithms but it is not the classifier model eventually deployed on the learning agent, we will not explain the entire model in the thesis (the model is briefly summarised in a pseudo description in Fig 4.6). Please see more details about how the model works in (Kong et al., 2013)

4.2.2 Data

FIGURE 4.7: Examples of Object Set for Multi-label Evaluation

The data used in this experiment is a collection of indoor visual objects from *Google Image*, as shown in Fig 4.7. We mainly separate this collection of images into two groups, 320 images for training different classification models (called S1 below) and the rest (160 images) for testing them. In this experiment, the visual features extracted from each image are used to train and test different visual classifiers. In order to eliminate the interference from noisy complex backgrounds, all objects are collected with a clean white background. All objects in the data are manually annotated by ourselves from two aspects, object class (e.g. apple, banana, book, mug, pen, stapler, wallet and glasses-case) and adjective colour attributes (e.g. black, blue, brown, green, pink, purple, red and yellow). We kept the numeric balance between different attributes (i.e. 60 instance for each attribute) when collected these objects.

Apart from the main object collection, we also build two extension versions of the S1 set (called S2 and S3 respectively as below), with additional training examples, to simulate an unbalanced situation that commonly occurs when continuously learning instances in real-time: 1) S2 set contains another 62 'mug' instances in the same colour besides images from the S1 set, and 2) S3 set adds extra 30 instance in each of "map" and "stapler" but in different colours based on S1. We propose to provide an unbalanced data in colour attributes in S2 and keep a relative balance in S3 respectively.

4.2.3 Experiment Procedure

In this experiment, we set up a task of learning real-object attributes, e.g. proper attributes (class name) and adjective attributes (colour), using multi-label classifier models. The task starts with small amount of training data (stage one), and then incrementally feed the models with more instances for particular objects/attributes (stage two). The latter stage

is to simulate a continuous learning process through dialogue with human tutors in the real world.

Here, in terms of the first stage of the learning task, we firstly train both visual classifiers (ML-kNN (Zhang and Zhou, 2007) and TRAM (Kong et al., 2013)) respectively with visual instance from the S1 training set, and evaluated with the rest images (160 images in total). It gives a basic view of how well they deal with numeric balanced learning examples in the visual-attribute learning task.

On the second stage, both classifiers will be continuously trained with additional visual instances (S2 and S3 individually), and also evaluated with the same test set. Since the balance of data is broken when incrementally learning more examples for specific attributes, evaluation results of this stage can show the truth about whether these algorithms can show constantly good performance in a continuous-learning process, which is more close to a real-time learning scenario with human tutors.

4.2.4 Metrics

To give a picture of the overall performance of two models, we report three values of each model, loosely following the evaluation metrics used by Kong et al. (2013):

- **Micro-F1 Score**: to measure the overall performance of a classifier model, considering both micro average of Precision and Recall with equal importance (i.e. the bigger value of F1-score presents a better performance)
- **Ranking Loss**: to evaluates the average fraction of predicted label-pairs that are not correctly ranked/ordered. An approach can achieve the best performance when its ranking loss is 0.
- **Average Prevision**: to evaluate the ranked labels, i.e. calculating the average fraction of labels ordered above a specific label l in the label set L, which actually belongs to L. (Note that the bigger its value is, the better performance a classifier model achieves.)

Moreover, to illustrate how well these approaches perform on each individual attribute, we plot the classification accuracy of each classification model with different training sets for evaluation.

On the other hand, since we focus on learning visual attribute with human tutors through real-time conversations, the agent is expected to response the user and also complete the

learning task as soon as possible through dialogue. We therefore consider the computational time of learning visual knowledge as another evaluation metric in this experiment. Here, we only consider the computational time of learning the whole training set.

4.2.5 Results & Discussion

Comparison of Approaches on Multi-label Classification Results of visual-attribute classification using two multi-label models (ML-kNN and TRAM) on S1 dataset are shown in the Table 4.2. In general, both models show much better performances on classification in proper attributes (class names) than in adjective attributes (colours). TRAM model with 0.206 in adjective attributes and 0.333 in proper attributes is performing better on Micro-F1 than ML-kNN model with 0.188 in adjective and 0.256 in proper. It might be explained that ML-kNN is a supervised learning model that requires much more training examples. On the evaluation of label ranking, TRAM also gets higher scores on average precision than ML-kNN, although it gets worse in ranking loss performance than ML-kNN, especially on proper attribute classification (i.e. 0.347 for ML-kNN and 0.661 for TRAM). The results suggested that TRAM model may work well on classifying attribute-based objects with little knowledge.

	Micro-F1		Ranking Loss		Average Precision	
Dataset	ML-kNN	TRAM	ML-kNN	TRAM	ML-kNN	TRAM
S1	0.188	0.206	0.461	0.451	0.187	0.213
S2	0.181	0.144	0.465	0.488	0.181	0.162
S3	0.125	0.125	0.503	0.464	0.125	0.147

a) Prediction of object colour

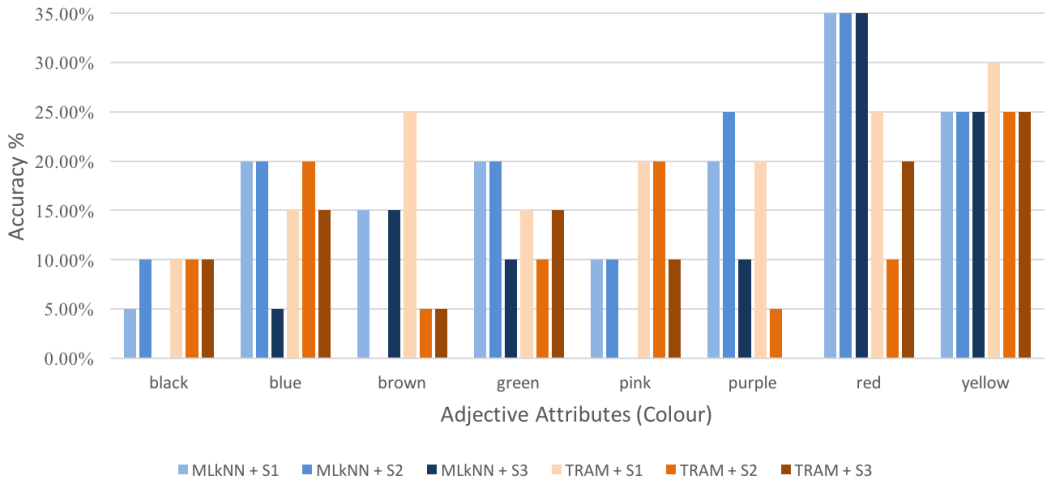
	Micro-F1		Ranking Loss		Average Precision	
Dataset	ML-kNN	TRAM	ML-kNN	TRAM	ML-kNN	TRAM
S1	0.256	0.333	0.347	0.661	0.2563	0.3235
S2	0.175	0.308	0.365	0.685	0.1750	0.2921
S3	0.231	0.308	0.373	0.681	0.2313	0.2880

b) Prediction of object type

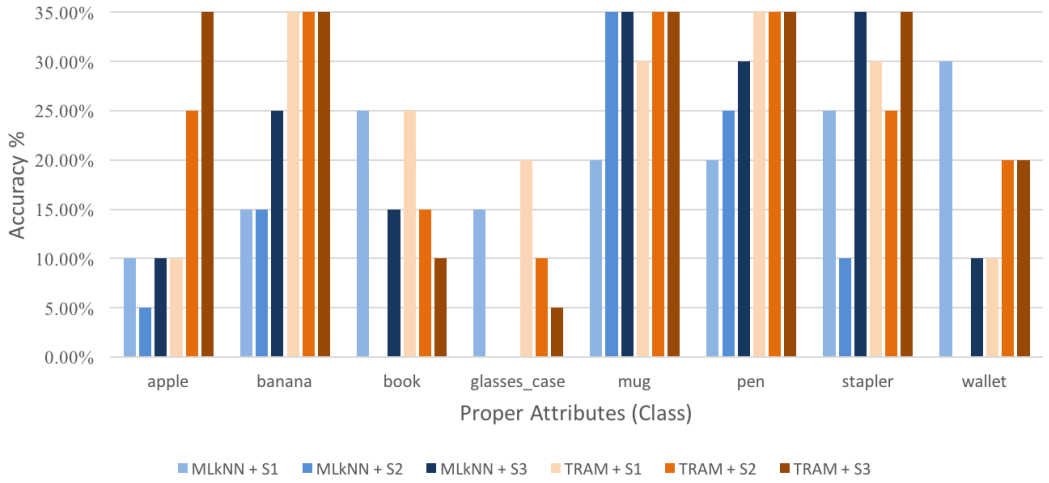
TABLE 4.2: Micro-F1, Ranking Loss and Average Precision on Prediction

Comparison of Approaches on a continuous learning process through dialogue

Table 4.2 and Figure 4.8 show how dialogue improve the performance of each model on S1, S2 and S3 datasets. Results in Table 4.2 indicate that both ML-kNN and TRAM models get worse in all attributes while the balance of data is broken by adding more instances for specific attribute, i.e. S2 and S3. Figure 4.8a demonstrates that the TRAM model manages to achieve a lower accuracy rate in adjective attributes. However, the ML-kNN model is more stable and accurate in the attributes "red" (about 35%) and "yellow" (around



(A) Predictions of colour attributes



(B) Predictions of object type labels

FIGURE 4.8: Accuracy of Prediction on Each Label

25%). In other adjective attributes, ML-kNN shows obvious fluctuation between S2 and S3, compared with the baseline dataset.

Furthermore, Figure 4.8b shows that TRAM model receives a set of higher accuracies than ML-kNN model in the proper attributes on the S1 dataset except the attribute "book". When there are more instances learned for specific attributes in classifiers on S2 and S3, both models have significant changes in the accuracy of each proper attribute. For S2 dataset, ML-kNN achieves more improvements (by nearly 55 percentage points) than TRAM (by about 45 points) in the attribute "mug", which has extra 62 instances learned. However, ML-kNN almost lost the capability of recognizing other attributes, especially "book", "glasses case" and "wallet", which accuracies are nearly 0%. In contrast, TRAM model keeps more stable and higher accuracies in almost all proper attributes, since ML-kNN performs as a majority classifier that predicts each object or attribute to the majority class with the most instances, especially when there is insufficient data trained. For S3 dataset, ML-kNN and

TRAM still show significant improvements in the attribute "mug" and "stapler" with more training instances. Meanwhile, TRAM also provide good and stable performances in almost all proper attribute. ML-kNN obtains better accuracies in most attribute compared with S2, since the data with extra 30 instances in "mug" and "stapler" keeps a relative balance in training data.

The results indicate that although it also leads to the drop of the accuracies in other attributes, an unsupervised learning model (i.e. TRAM) may contribute to alleviation of the unbalanced data in such continuous learning process.

Comparison of Approaches on Computational Time Regarding with the time cost on training classifier, table 4.3 shows the transductive model TRAM (13.87s) spend nearly triple as much time as the ML-kNN model (4.01s) spend on the baseline dataset. The training time by both models gradually rise with the increasing number of training instances.

Classifier Model	S1 Set	S2 Set	S3 Set
ML-kNN	**4.01s**	**4.68s**	**4.94s**
TRAM	13.87s	13.54s	13.95s

TABLE 4.3: Computational Time on multi-label learning, while learning with different image collections (S1, S2, S3)

Time cost (or dialogue cost) in a teachable system is used to measure how fast a system can learn to identify novel objects or attributes (Walker et al., 1997). The less model cost on the training process, the better the performance will be. Hence, the ML-kNN model shows a higher speed than the TRAM model with three different datasets.

4.2.6 Summary

Following the experiment results above, neither ML-kNN nor TRAM unfortunately can address the attribute learning task in a continuous learning task, i.e. they can achieve better classification performance but with longer computational time, or vice versa.

In the next experiment, we will keep exploring the most appropriate classifier method for the interactive learning task with humans. Instead of testing multi- or single-label approaches, we will focus on investigating the efficiency of an incremental method on such learning task.

4.3 Experiment 2: Learning Visual-Attribute from Dynamic Training Data

The experiment presented in this section aims at exploring an appropriate method that can support visual-attribute learning/classification incrementally, with dynamic training examples, instead of static data. We compare the chosen method with several baseline approaches, including one of above-mentioned multi-label classification methods (ML-kNN (Zhang and Zhou, 2007)).

4.3.1 Classification Approach

In this section, we choose to implement a simple online classification model – logistic regression SVM with Stochastic Gradient Descent (SGD-SVM) – to incrementally create and update a binary classifier per attribute while learning single unseen object.

4.3.1.1 SGD-SVM

The SGD-SVM classification method is a simple but very efficient method that discriminatingly learns linear classifiers under convex loss functions, e.g. *logistic regression*, optimised using the *Stochastic Gradient Descent (SGD)* algorithm.

SGD Algorithm In order to improve the learning efficiency, the Stochastic Gradient Descent (SGD) algorithm is applied to learn examples based on a single example z_t instead of the gradient of an empirical risk $E_n(f)$ (Bottou, 2012), as below:

$$w_{t+1} = w_t - \lambda_t \bigtriangledown_w Q(z_t, w_t), \tag{4.9}$$

where the stochastic process $\{w_t, t = 1....\}$ relies on the training samples randomly selected at each iteration t. It allows the learning method to process examples as quickly as possible, without remembering any training samples visited previously (considered as an incremental process). In this case, SGD algorithm is likely to directly optimise the expect risk $E(f)$ that measures the learning procedure performance of the future examples (Bottou, 2012).

Logistic Regression The logistic regression function, as an S-shaped function ranged between 0 and 1, models probabilities for a specify classes/labels, where an output belongs to a certain class c if its probability approaches to 1, otherwise not. Given the SGD algorithm, it aims at minimizing the cost function of the logistic regression approach indirectly using SGD algorithm (also known as a online Gradient Descent (OGD)), as shown in Eq. 4.10.

$$p(x_i \mid y) = \sigma(w^T x + b), where\ w \leftarrow w - \lambda_t = \begin{cases} \lambda_w, & \text{if} y_t w^T \Phi(x_t) > 1, \\ \lambda_w - y_t \Phi(x_t), & \text{otherwise} \end{cases} \quad (4.10)$$

where σ represents a logistic function, $\Phi(x_t)$ represents a set of features for example x_t and λ is defined as a hyper-parameter for controlling the loss function.

In this thesis, we implement this incremental SGD-SVM classifier using a powerful Machine Learning tool-kit (WEKA (Frank et al., 2010)).

4.3.2 Data

In this experiment, we make the use of a benchmark dataset of natural object-based images with attribute annotations – the aPascal-aYahoo data set[4]. This data set has two subsets: the Pascal VOC 2008 dataset and the aYahoo dataset. The Pascal VOC 2008 dataset is created for visual object classifications and detections. The aPascal data set covers 20 attribute-labelled classes and each class contains a number of samples, ranging from 150 to 1000. The aYahoo dataset, as a supplement of the aPascal dataset, contains objects similar to aPascal, but with different correlations between attributes. It only contains 12 object classes. Images in dataset are annotated with 64 binary attributes, covering shape and material as well as object components (see table 4.4). We use the 6340 images selected by Farhadi et al. (2009) from the aPascal dataset for training and use the whole aYahoo dataset with 2644 images as the test set. As both aPascal and aYahoo data sets are imbalanced in the number of positive instances for each attribute, as shown in table 4.4, this might affect the performance of the models on attribute classification.

4.3.3 Experiment Procedure

In this experiment, we compare the SGD-SVM incremental classifier with another two approaches, L2-loss Linear SVM (Farhadi et al., 2009) and ML-kNN model (Zhang and Zhou, 2007). We train and test each classification model with the aPascal-aYahoo dataset. However, different with the previous experiment, instead of training with the whole dataset at once, classifiers are trained with each single visual instance one-by-one, similar to a real-time learning process. We run a 20-fold cross validation experiment, each fold of experiment will present the visual instances in a random sequence. It means that the classifiers will be

[4] http://vision.cs.uiuc.edu/attributes/

Attribute	aPascal	aYahoo	Attribute	aPascal	aYahoo	Attribute	aPascal	aYahoo
2D Boxy	207	146	3D Boxy	393	752	Round	39	179
Vert Cyl	195	334	Horiz Cyl	94	286	Occluded	1913	778
Tail	184	529	Head	1737	1157	Ear	1097	1048
Snout	237	708	Nose	995	345	Mouth	930	332
Hair	1095	216	Face	1022	392	Eye	1183	1061
Torso	1538	1024	Hand	811	364	Arm	1080	383
Leg	994	922	Foot/Shoe	604	719	Wing	114	11
Window	304	167	Row Wind	86	224	Wheel	336	64
Door	192	13	Headlight	162	36	Taillight	104	5
Side mirror	150	71	Exhaust	50	41	Handlebars	92	37
Engine	35	71	Text	84	388	Horn	4	145
Rein	32	284	Saddle	20	121	Skin	1396	161
Metal	581	739	Plastic	260	459	Wood	195	167
Cloth	1591	123	Furry	250	996	Glass	180	34
Feather	99	1	Wool	12	15	Clear	32	42
Shiny	432	527	Leather	6	85			

TABLE 4.4: The Number of Positive instances on each attribute in aPascal-aYahoo Datasets (aPascal for training set, aYahoo for testing Set, attributes with no testing instances removed)

trained with different visual instances at the same time stamp, although all of them have the same training data.

In order to investigate how well these model performs with fewer training examples, we will pause the training process and evaluate the existing classifiers with the aYahoo dataset, when the classifiers has seen half of the training examples from the aPascal data. We will also evaluate these classifiers at the end of learning process with learning the full set of training samples.

4.3.4 Metrics

The metrics for this experiment are similar for the previous one (Experiment 1), i.e. we plot the classification accuracy of these approaches for each visual attribute, and report and compare the computational time of each method. Different with the previous experiment, we plot the curve of how fast the computational time of each classifier model increases after learning each single instance.

On the other hand, we report the average F1-score, Accuracy and Area under ROC curve (AUC) (an estimate of how well a classifier model can predict a random (positive) example as positive (called True positive), in comparison to a random negative example as negative (True Negative)) across all classifiers, over all the objects.

4.3.5 Results & Discussion

Comparison of Approaches on Recognition Performance The results (see Fig 4.9) show that while the models sometimes perform quite poor on specific attributes (such as the attributes '3D Boxy', 'Furry' and 'Occluded'), the performance over all attributes in general is good. We note that the shapes of the plots using ML-kNN and SGD-SVM are very similar, both algorithms show worse performance on some attributes that are not generally distinctive (easy to detect). For example, the attribute 'Occluded' with 1913 training instances is performing relatively worse (less than 70% accuracy) using both algorithms.

Compared with other methods (ML-kNN and SGD-SVM), the linear-SVM classifier model is performing far worse on most of attributes, especially on the attribute 'Horn', this might be because the features points of those attributes are not linearly separable.

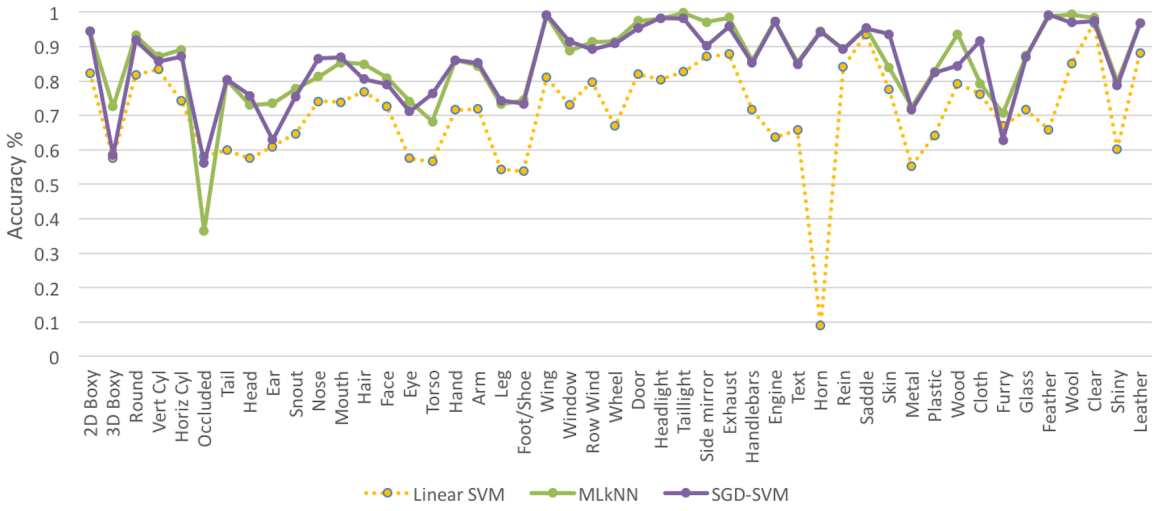


FIGURE 4.9: Accuracy on each attribute for each method (SGD-SVM, ML-kNN and Linear-SVM)

In order to investigate the overall performance of three models with different number of training samples, we also report and compare the average scores (including *Micro-F1*, *Accuracy* and AUC) for each method, computed across all of the attributes, of models learned with the **whole** set of training examples (Table 4.5(a)) and the **half** training set (Table 4.5(b)) respectively.

The results in Table 4.5 has demonstrated that SGD-SVM model is generally performing better than other classifier approaches across all of the attributes in this experiment, no matter learning with full or half of the training set. Specificity, given the model learned with all training examples (see Table 4.5(a)), SGD-SVM achieves comparable scores on both Accuracy (0.858) and AUC (0.631) with the ML-kNN model, although its Micro-F1 is slight lower than other models. In terms of the situation with half training set (Table

Model	mean Micro-F1	mean Accuracy	mean AUC
Linear-SVM	0.218	0.711	0.545
ML-kNN	0.200	0.856	**0.631**
SGD-SVM	0.191	**0.858**	**0.631**

a) models rained with the *whole* set of training examples

Model	mean Micro-F1	mean Accuracy	mean AUC
Linear-SVM	0.221	0.679	0.544
ML-kNN	0.168	0.804	0.610
SGD-SVM	**0.237**	**0.817**	**0.623**

b) models trained with the *half* set of training examples

TABLE 4.5: Overall Performance of Three classifier Models (Linear-SVM, ML-kNN, SGD-SVM) on predicting Attributes

4.5(b)), although the overall performance of three models are generally decreased because of fewer examples, the SGD-SVM model still outperforms others on all measures, i.e. 0.237 on Micro-F1, 0.817 on Accuracy and 0.623 on AUC. We can therefore conclude that, in terms of recognition performance, the SGD-SVM model is more desirable than other models in the visual-attribute learning task.

Comparison on Computational Time Apart from the recognition performance presented above, we also consider how fast a classification method can learn a single novel object/attribute through real dialogue with humans. Given that an learning visual attributes is considered in a real-time learning task, higher computation times is usually unacceptable. Fig. 4.10 plot how the computational of each model changed across all attributes with learning all training examples (i.e. 6340 instances in total). Note that in this research, we are more concerned with whether the classifier can quickly learn all attributes for a single instance. Hence, we record the time cost of each method with each single object, instead of a particular attribute. We define a **Learning Step** as comprised of 100 training instances. The time cost will be recorded at the end of each learning step.

Figure 4.10 shows that the SGD-SVM is more appropriate for an interactive learning task than other approaches, because it is considerably faster (only 0.67s in average) than the others (around 200s in average). Although the ML-kNN model shows a comparable performance in the first 2300 instances with the SGD-SVM model, its cost significantly increases afterwards. This might be because these learning methods (linear-SVM and ML-kNN) are normally required to retrain them with the entire set of training samples seen so far whenever a new instance is provided. However, the SGD-SVM classifier, as an incremental learning method, can update its model without re-computing previous increments. This capability of the SGD-SVM classifier may lead to a quicker response than other methods in NL interaction, especially with a large amount of training examples.

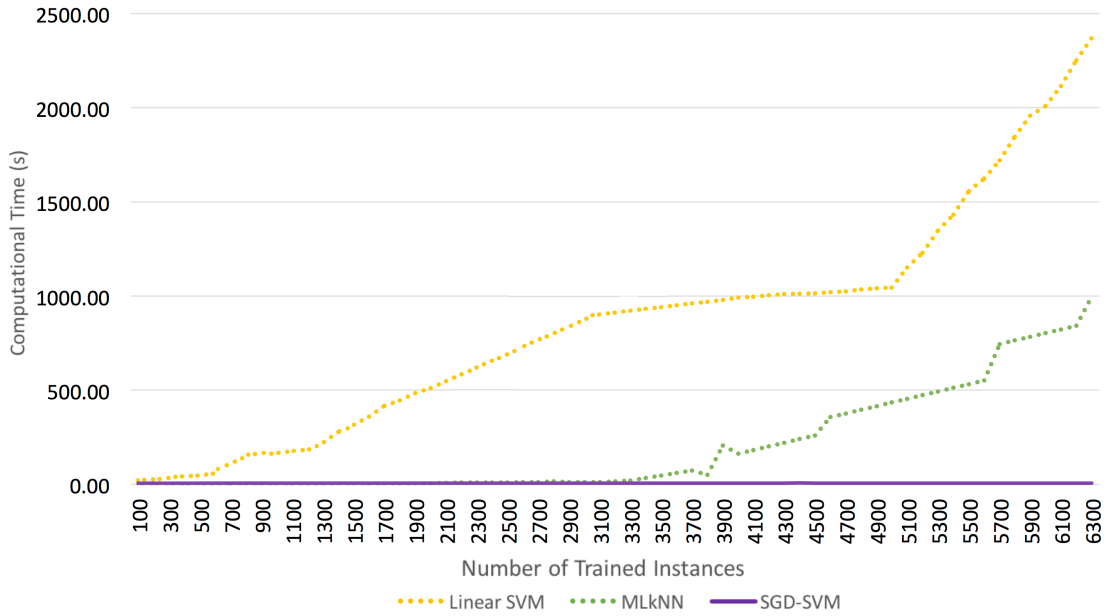


FIGURE 4.10: Time Consumption on learning each instance for each method (SGD-SVM, ML-kNN and Linear-SVM)

4.3.6 Summary

The results above have suggested that the SGD-SVM classifier model is more desirable while learning new information in real time, i.e. it achieves better performance with less time cost of training than others. In this thesis, we train a set of binary SGD-SVM classifiers, each for one visual attributes (e.g. "red", "blue", "square" and etc.). Keeping these attributes separate can support the agent to easily extend to new visual attributes or attribute categories (e.g. texture), i.e. while learning a new classifier for a specific attribute that the agent has never seen before, the recognition performance on other classifiers will not be affected since there is nothing dramatically changed with their classifiers.

4.4 Chapter Summary

In this chapter, we briefly reviewed a list of existing classification approaches on two core dimensions: 1) *single-* or *multi-label* learning problem, and 2) learning from static (*offline*) or dynamic training data (*online*). Among these approaches, we attempted to select and compare some exiting classifier models on learning novel objects and their attributes in two experiment, i.e. learning visual attributes in the multi-label learning task and with in a real-time incremental learning process. The results obtained from experiments suggest that:

1. *None of the multi-label classifier models can achieve good performance on the visual attribute classification (higher accuracy with less training time)*

2. Neither multi-label classification models nor conventional linear SVM methods can address the real-time learning task, in which the model is required to operate online and learn incrementally and interactively, because they requires retaining with all training data with every single instance.

We have shown that the incremental **SGD-SVM** model achieved comparable performance with higher accuracy on both a full and half training set. This approach outperforms other classification methods as it is not only able to accurately predict visual attributes, but also update models faster while learning new visual instances. In this project, we will therefore employ the incremental SGD-SVM classifiers as key component within the proposed multi-modal framework (see Chapter 3) [5].

For the aforementioned reasons, this thesis is concerned about an incremental learning process through real-time, natural language interaction with real human tutors. Given a more complicate interaction situation with massive variation and uncertainties, interacting with humans can always provide further help for learning efficiency. However, prior to the dialogue processing in this task, we attempt to investigate how humans talk with each other for teaching/learning specific visual objects. We are concerned with this problem in two key questions: 1) *what dialogue capabilities and strategies does the tutor or the learner perform to gain information from each other?* and 2) *what special dialogue phenomena are involved into this human-human conversations given the learning task?* In the next chapter, we collect a set of realistic human-human conversations on an interactive visual-attribute learning task, which will help answer those questions.

[5] As part of this research, the latest system is extended to interactively learn real object classes, instead of visual attributes, by integrating a deep learning based classification approach (an adapted version of a Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) (Zhang et al., 2014) and a pre-trained Convolutional Neural Network based on the architecture proposed in Krizhevsky et al. (2012): we take the output of the final layer in the CNN model (attribute representation) as input to incrementally learn object classes with the LB-SOINN model). Although it shows good performance on the learning accuracy, it still requires a large amount of training data (i.e. around 600 images per visual instance) to learn each single object. In addition, updating the classification model with each single example might take more than 1s.

Chapter 5

BURCHAK: Human-Human Dialogue Corpus for Learning Visual-Attribute

In order to build an interactive learning agent for the visual language grounding/learning task, we make an effort on collecting natural conversations about learning novel visual attributes (for example, colour and shape) with grounding by demonstration. We create a new freely available dialogue corpus on the task of learning novel visual attributes, called BURCHAK, in which the learner is required to cooperate with his fellow tutor to explore/learn unknown visual attribute words by talking about different objects within each dialogue. To our knowledge, this corpus is the first dataset that addresses such multi-modal learning problems through natural, spontaneous dialogue. It shows naturalness on two key aspects: 1) task-oriented dialogue actions, i.e. given the attribute learning task, the participant behaves naturally like talking within the real world and 2) linguistic/dialogue phenomena, i.e. it produces some phenomena that usually appear as 'noise' or 'performance errors' in a naturalistic dialogue context, for example, self -repair and -repetition, filler, and etc..

Crocker et al. (2000), Ferreira (1996), Purver et al. (2009b) argued that natural, spontaneous dialogue is *inherently incremental*, and thus gives rise to dialogue phenomena such as self- and other-corrections, continuations, unfinished sentences, interruptions and overlaps, hedges, pauses and fillers. As we note in Yu et al. (2017b), these phenomena are interactionally and semantically consequential and contribute directly to how dialogue partners coordinate their actions and the emergent semantic content of their conversation. They also strongly mediate how a conversational agent might adapt to their partner over time. For example, self-interruption, and subsequent self-correction (see e.g. Table. 5.4) as well as hesitations/fillers (see e.g. Table. 5.7) are not simply noise and are used by listeners to guide

linguistic processing (Clark and Fox Tree, 2002); similarly, interruptions and subsequent continuations (see e.g. Table. 5.6) are performed deliberately by speakers to demonstrate strong levels of understanding (Clark, 1996).

Despite this importance, these phenomena are excluded in many dialogue corpora, and glossed over/removed by state of the art speech recognisers (e.g. Sphinx-4 (Walker et al., 2004) and Google's web-based ASR (Schalkwyk et al., 2010); see (Baumann et al., 2016) for a comparison). One reason for this is that naturalistic spoken interaction is excessively expensive and time-consuming to transcribe and annotate on a level of granularity fine-grained enough to reflect the strict time-linear nature of these phenomena.

We, therefore, collect those natural conversations using a new *incremental* variant of the DiET chat-tool (Healey et al., 2003, Mills and Healey, 2017), which enables a character-by-character, text-based interaction between pairs of participants, and which circumvents all transcription effort as all this data, including all timing information at the character level is automatically recorded.

In this chapter, we look at a list of existing human-human dialogue corpora (in Section 5.1), which are mainly discussed on three dimensions: 1) contains either written or spoken dialogues, 2) whether conversations are taking place on the visual-attribute learning task, and 3) whether it contains dialogue phenomena in incremental language processing, e.g. self -repetition and -repair, continuation and hesitation. It is followed by a description of a new visual-attribute learning task applied in the data collection experiment (Section 5.2). In the Section 5.3, we further statistically analyse the collected BURCHAK corpus on the aspects of dialogue capability, strategies and naturalistic phenomena (as mentioned above). This section also describes a set of further corpus processes, including the cleaned-up procedure and annotation protocol. Finally, we briefly make a discussion (Section 5.4) on the similarities and differences between text chat and spoken iteration given the interactive visual-attribute learning task.

5.1 Human-Human Dialogue Corpora

In the past decades, there are a large number of dialogue corpora created for either studying the human behaviours in the conversation or learning a robust dialogue system. One of the most essential distinctions between the existing dialogue corpus is whether the dialogue involves interaction between a human and a machine, or only between humans. In this thesis, we are more interested in the human-human dialogue corpora than human-machine dialogues, as they reflect more natural dialogue interaction with more variations and unexpected phenomena which cannot be obvious in the human-machine dataset. This section

describes and discusses a list of existing dialogue corpora involving conversations between humans. These corpora can be generally categorised into two groups, e.g. spoken dialogues (Section 5.1.1) and written dialogues (Section 5.1.2). Table. 5.1 summaries a list of existing Human-Human Dialogue corpora on a variety of topics.

Corpus	Type	Topic	Total # of dialogues	Incremental Phenomena
HCRC MAP TASK (Anderson et al., 1991)	Spoken	navigation	60	✓
The Switchboard (Godfrey et al., 1992)	Spoken	Casual Topics	2,400	✓
The British National Corpus (Leech, 1992)	Spoken	Casual Topics	854	✓
The Settlers of Catan Corpus (Afantenos et al., 2012)	Written	Game terms	21	
The Walking Around Corpus (Brennan et al., 2013)	Spoken	navigation	36	✓
Cardiff Conversation Database (Aubrey et al., 2013)	Spoken	Unrestricted	30	
The Dialogue State Tracking Challenge 4 (Kim et al., 2016)	Spoken	hotels, flights & car rentals	35	
The Ubuntu Dialogue Corpora (Lowe et al., 2015)	Written	Ubuntu Operating System	930K	
Reddit	Written	Unrestricted	-	

TABLE 5.1: Existing Human-Human Dialogue Corpora

5.1.1 Human-Human Spoken Corpora

We firstly review several dialogue corpora in which human participants communicate with each other over the phone or face-to-face. Serban et al. (2015) pointed out that, compared with written dialogues, spoken conversations are more inform, use shorter words/phrases, because the speaker is talking in a track-of-thought manner. In this section, we describes and discusses the spoken dialogue corpora in two groups respectively: non-task-oriented and task-oriented dialogues.

5.1.1.1 Non-task-oriented Spoken Corpora

Some corpora, such as (Aubrey et al., 2013, Godfrey et al., 1992, Leech, 1992), collect a number of conversations which topics are either not pre-specified or casual. These corpora are defined as spontaneous corpora Serban et al. (2015), in which they closely resemble spontaneous and unplanned spoken conversations between humans.

The Switchboard (Godfrey et al., 1992), as one of the most influential spoken dialogue corpus, is a large-scale multi-speaker dialogue corpus, consisting of approximately 2,500 dialogues in the form of both conversational speech and text. These conversations were collected with 500 speakers using a telephone-based application. All conversations are transcribed at the word level, i.e. it is time aligned at the word level using a supervised phone-based

recognition. Each conversation involves into a list of social dialogue phenomena, e.g. speakers turns, simultaneous talking, interrupted sentences, partial words and other phenomena commonly occurs in other conversational speech.

The British National Corpus (BNC) (Leech, 1992) is another essential dialogue dataset consisting of over 10 million words of dialogue. Dialogues in this corpus covers a wide range of topics from business/government meetings to radio shows as well as phone-ins. All dialogues are transcribed at the lexical level. Leech (1992) applied a grammatical-tagging form – part-of-speech (POS) – to tag each word in the corpus. Those conversations also contains a number of incremental dialogue phenomena, such as split utterances, interruptions, self-repetition and -corrections.

Aubrey et al. (2013) introduced an audio-visual database called Cardiff Conversation Database (CCDb), which consists of 30 unscripted dialogues between pairs of speakers. Each dialogue was collected in a 5-min conversation. The corpus records, not only the speech, but also the facial expressions and gestures for each pair of participants. The role of the participants (either speaker or listener) was not defined before the conversation. In the original dataset(Aubrey et al., 2013), "eight dialogues are fully annotated for speaker activity, facial expressions, head motion, and non-verbal utterances". This corpus shows some basic dialogue phenomena, e.g. front-channel and back-channel.

5.1.1.2 Task-oriented Spoken Corpora

Here, we describe some dialogue corpora on a specific topic or a task of solving particular problems, e.g. (Anderson et al., 1991). They are applied to build goal-oriented dialogue systems on specific domains.

HCRC MAP TASK (Anderson et al., 1991), as a well-known data set, consists of 60 unscripted, task-oriented dialogues on the task of cooperative problem solving. Each pair of participants were required to collaboratively reproduce the route on the map through face-to-face conversations in a recording studio. The corpus contains around 150,000 words. It also explores the performance of human speech by controlling the familiarity and eye-contact between speakers. Similar to this corpus, we also explore the effects of several independent variables on the quality of human-human conversations (see Section 5.3.3). All dialogues in this corpus are transcribed at the orthographic level, including filled pauses, false starts and repetitions, broken words and also overlapped regions. Anderson et al. (1991) also provide annotations for dialogues, like abandoned words, letter names, filled pauses as well as editorial uncertainty.

Similar to the HCRC MAP TASK corpus, the Walking Around corpus (Brennan et al., 2013) is another direction-giving task-oriented dialogue corpora, which consists of a small number of dialogues (only 36) between pairs of participants over mobile phone. The task involves two steps: 1) a 'stationary partner', equipped with a map, directs a 'mobile partner' to find 18 destinations on the university campus to take photos; and 2) participants are required to duplicate the photos taken by the 'mobile partner' via communicate in person. These conversations are transcribed with timestamps. Each conversation covers a list of dialogue phenomena, e.g. self-correction, self-repetition and overlapping.

The Dialogue State Tracking Challenge (DSTC) 4 (Kim et al., 2016) is a small conversation corpus, which contains only 35 dialogues, with over 30 thousand utterances and around 273,000 words. These conversations are collected through interaction between tourists and tour guiders over Skype, where they were discussing a list of particular topics, such as hotels, flights, as well as car rentals. These task-oriented dialogues show a range of turn-taking phenomena, with well-formed and clear conversation structure. All conversations are manually transcribed into JSON-formatted logs with annotations at the speech act, slot-filling and semantic levels (see example in Table 5.2). All annotations are provided at the utterance level.

INPUT :	TOURIST: Can you give me some uh- tell me some cheap rate hotels, because I'm planning just to leave my bags there and go somewhere take some pictures.
OUTPUT :	-
INPUT :	GUIDE: Okay. I'm going to recommend firstly you want to have a backpack type of hotel, right?
OUTPUT :	*SLOTS: <DET CAT="MAIN">backpack type </DET > <LOC FROM-TO="NONE" REF="HOTEL "CAT="HOTEL">hotel</LOC > SPEECH_ACTS: FOL (ACK) , INI (RECOMMEND) , QST (PREFERENCE)*
INPUT :	TOURIST: Yes. I'm just gonna bring my backpack and my buddy with me. So I'm kinda looking for a hotel that is not that expensive. Just gonna leave our things there and, you know, stay out the whole day.
OUTPUT :	-
INPUT :	GUIDE: Okay. Let me get you hm hm. So you don't mind if it's a bit uh not so roomy like hotel because you just back to sleep.
OUTPUT :	*SLOTS:<DET CAT="MAIN">not so roomy</DET > SPEECH_ACTS: FOL (ACK) , QST (PREFERENCE)*

TABLE 5.2: Annotation Example in DSTC4 Corpora (Kim et al., 2016)

5.1.2 Human-Human Written Corpora

In contrast to those human-human spoken corpora above, although these written conversations contains more well-formed and complex utterances than spoken dialogues, it easily contains some mistakes (e.g. spelling errors) and abbreviations, both of which are not

recorded in spoken dialogues. The written corpora can be classified into three groups based on the corpora format, including 1) form: contains dialogues from the forum-based website (e.g. Reddit[1]) in which users are able to make posts and reply/discuss other's posts via comments; 2) micro-blogging: conversations on the micro-blogging website (like Twitter or Facebook) requires shorter utterances in each message; 3) chat-based: all conversations occurs in the real-time between users. Here, we mainly focuses on a set of chat-based corpora that in which humans chat with each other using a text-based chat tool (e.g. Facebook messager). We will review two important task-specified dialogue corpora: The Settlers of Catan Corpus (Afantenos et al., 2012) as well as Ubuntu Dialogue Corpus (Lowe et al., 2015)

The Settlers of Catan Corpus, created by Afantenos et al. (2012), is a pilot dialogue dataset containing logs from 40 games with approximately 80,000 annotated utterances over an online game league[2]. These conversations have been concerned about bargaining, strategic negotiations in the game of 'The Settlers of Catan'. The majority of these turns in the corpus involve offers, counteroffers, and acceptances or rejections of offers. Afantenos et al. (2012) provided a multi-layer annotation schema for those conversations from relatively simple to complex: 1) determines the addressee, annotates the speech-act of each elementary discourse unit (EDU), as well as tags the strategic play which provides more information about strategic reasoning; 2) identifies the discourse structure and relations between EDUs; 3) annotates the preferences computed from the domain-level actions.

The Ubuntu Dialogue Corpora (Lowe et al., 2015) is a large dialogue data set created from the Ubuntu Chat Logs. It consists of approximately 1 million task-specific conversations, with over 7 million utterances and 100 million words. Given four tuples of time, sender, recipient as well as utterance for each message, they extract all conversations between two humans. Lowe et al. (2015) announce that this corpus only take into account conversations which have more than three turns.

5.1.3 Summary

Here, we reviewed a set of corpora collecting realistic conversations between humans, none of which however can fit into our domain, suitable for training multi-modal conversational agents that perform the task of *actively learning visual concepts* from a human partner in *natural, spontaneous* dialogue. Some corpus, such as SWITCHBOARD (Godfrey et al., 1992), the British National Corpus (Leech, 1992), MAPTASK Thompson et al. (1993) and

[1] http://www.reddit.com
[2] http://settlers.inf.ed.ac.uk/

the Walking Around corpora (Brennan et al., 2013), contain many of the incremental dialogue phenomena that we are interested in here, but there is no shared visual scene between participants, meaning we cannot use such data to explore learning of perceptually grounded language. The grounded word meanings are not taught by ostensive definition as is the case we considered in this task – interactively learning visual colour and shape attributes.

Recently, a multi-modal question-answer (QA) corpus by Das et al. (2016) is introduced as a large-scale dataset on the Visual Dialogue task, with 1200,000 QA pairs, i.e. 1 dialogue with 10 question-answer pairs on total 120k images from COCO image set (Lin et al., 2014). However, this corpus dose not take into account natural, spontaneous conversations (dialogue phenomena) in the collection procedure.

In this chapter, we therefore design an experiment to collect human-human dialogues for a novel visual-concept learning task (as shown in the following sections). To our knowledge, this is the first corpus considering incremental phenomena in this learning domain.

5.2 The Method: a Shape and Colour Learning Task

In this experiment, we introduce a visual attribute learning task to simulate a natural conversation between the tutor and the learner on learning the visual colour and shape of a certain object. Different to the previous work that commonly applies a Wizard-of-Oz (Woz) technique (Fraser and Gilbert, 1991) to investigate how humans interact with a robot/machine, Riek (2012) and Li and Dey (2013) emphasised that one of the difficulties in the Woz method is that some features in the human behaviour, e.g. complex decision-making and unexpected errors or mistakes, cannot be easily simulated by the Wizard (machine). In this chapter, we, therefore, design a human-human conversation data collection given a visual learning task. It brings more benefits on the investigation of realistic human behaviours in both roles of participants, the tutor and the learner. It allows us to investigate, not only about how people would teach novel knowledge but also about how humans as the learner can acquire useful information, through natural conversations.

In this task, the participants will be randomly assigned to two experimental roles: the *Tutor* versus the *Learner*. The learner is required to cooperate with his fellow tutor to explore 6 visual attributes, by talking about visual attributes (e.g. colour and shape) through a sequence of 9 visual objects, one at a time. These objects are created based on a 3 x 3 visual attribute matrix (including 3 colours and 3 shapes (see Fig.5.1e)).

We design the task in this way to collect data for situations where a robot has to learn the meaning of human visual attribute terms. In such a setting the robot has to learn the

perceptual groundings of words such as "red". However, in the task, since all participants, especially who are in the role of learner, have been equipped with rich background knowledge about the visual groundings (i.e. they have known what "red" looks like and what is the meaning of "square" before the experiment), in order to collect data about teaching/learning such perceptual meanings, we invented new attribute terms whose groundings the Learner must discover through interaction: we assigned the visual attributes (in Fig.5.1) to new unknown words in a made-up language, instead of standard English words, for instance, "sako" for red and "burchak" for square. In order to make it easier to pronounce and remember by both native and non-native speakers, we invent these new words based on Japanese pronunciation. During the experiment task, participants are not allowed to use any of the usual colour and shape words from the English language (see Appendix B.2 and B.3)): the participant, who talks about the visual attribute using standard English words, will be penalised by deducting the final score, which is applies to pick the winner in a competition mechanism (see more details as below).

Procedure In the experiment, both the learner and the tutor were given written instructions about the task and had an opportunity to ask questions about the procedure (detained in Appendix B). They were then seated back-to-back in the same room, each at a desk with a PC displaying the appropriate task window and chat client window (see Fig.5.1). They were asked to go through all visual objects in at most 30 minutes and then the Learner was assessed to check how many new colour and shape words they had learned at the end of each experiment (Appendix B.4).

Competition Mechanism To encourage the participants to memories the visual attributes in such interactive learning task, we introduced a competition mechanism: picking the winner (the best performing pair) among all pairs of participants by comparing their final scores. The final score (see Appendix B.2 and B.3)) is comprised of three core parts:

1. **Scores from Learners' Assessment**: As mentioned above, the learner will be tested for the number of visual-attribute words the learned learned from conversations, each correct word counts 1. The assessment result is calculated by the number of correct words.

2. **Scores of Learners' memory**: This score take into account how many times the learner can infer the attribute name without any hints by the tutor. During the teaching/learning conversation, it counts 1 every time when the learner gives correct attribute words (either colour or shape) before the tutor in the beginning of dialogues with new object.

3. **Penalties**: there are two kinds of penalties applied in the experiment. One of them is the penalty for the forbidden words, i.e. the final score will be deducted every time

when the participant attempts to describe the visual attributes using standard English words, penalised by −0.5 point. Another penalty is applied for overtime work, i.e. if a pair of participants cannot pass through all 9 visual objects in 30 minutes, their final score will be penalised with −5.

At the end of the entire experiment, except the participation voucher (£10), the best performing pair was also given a £20 Amazon Voucher as prize.

Following this task formula, we collect the data using a novel, character-by-character variant of *the DiET chat tool* (Healey et al., 2003, Mills and Healey, 2017). The chat-tool by Healey et al. (2003), Mills and Healey (2017) is designed to support, elicit, and record at a fine-grained level, dialogues that resemble face-to-face dialogue in that turns are: (1) constructed and displayed incrementally as they are typed, (2) transient, (3) potentially overlapping, and (4) not editable (i.e. deletion is not permitted), as describe below.

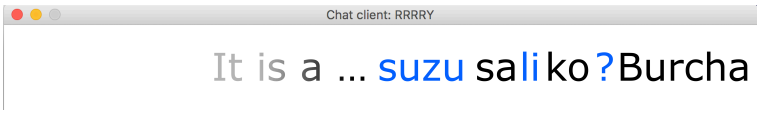
5.2.1 The DiET Experimental Toolkit

This is a custom-built Java application (Healey et al., 2003, Mills and Healey, 2017) that allows two or more participants to communicate in a shared chat window. It supports live, fine-grained and highly local experimental manipulations of ongoing human-human conversation (see e.g. Eshghi and Healey (2015)). The variant we use here supports text-based, character-by-character, interaction between pairs of participants, and here we use it solely for data-collection, where everything that the participants type to each other passes through the DiET server, which transmits the utterance to the other clients on the character level and all are displayed *on the same row/track* in the chat window (see Fig.5.1b) - this means that when participants type at the same time in interruptions and turn overlaps, their utterances will be all jumbled up (see Fig. 5.1b). To simulate the transience of speech in face-to-face conversation with its characteristic phenomena, all utterances in the chat window fade out after 1 second (see Fig. 5.1b, c and d). Furthermore, like in speech, deletes are not permitted: if a character is typed, it cannot be deleted.

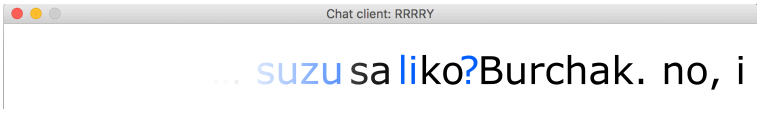
Since the learner is expected to identify the shape and colour of the presented objects correctly for as many objects as possible, the tutor initially needs to teach the learner about these using the presented objects. For this, the tutor is provided with a visual dictionary of the (invented) colour and shape terms (see Fig. 5.1e), but the learner only ever sees the object itself. The learner will thus gradually learn these and be able to identify them, so that initiative in the conversation tends to be reversed on later objects, with the learner making guesses and the tutor either confirming these or correcting them.

T(utor): it is a ... [[sako]] burchak.
L(earner): [[suzuli?]]
T: no, it's sako
L: okay, i see.

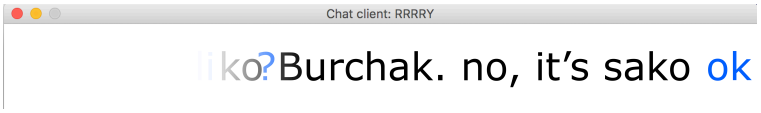
(a) Dialogue Example from the corpus



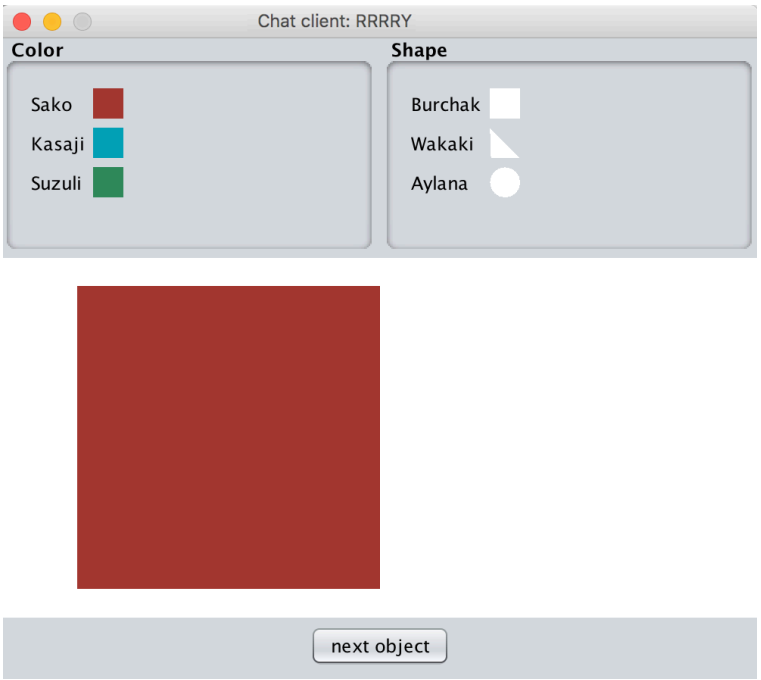
(b) The Chat Tool Window during dialogue in (a) above



(c) The Chat Client Window after 3 seconds



(d) The Chat Client Window after 6 seconds



(e) Task Panel for Tutor (Learner only sees the object)

FIGURE 5.1: Snapshots of a conversation example through the DiET Chat tool, where two participants are talking about the visual attributes of the certain object in (e). In the client windows (b,c,d), black characters represent one participant and the blue characters are typed in by another participant. ('sako' is the invented word for 'red', 'suzuli' for green and 'burchak' for square)

5.2.2 Participants

Forty participants are recruited from among students and research staff from various disciplines at Heriot-Watt University. The experiment design takes into account some independent variables, including:

- Native Language – includes 22 English native speakers and 18 non-native speakers.
- Familiarity – includes 13 pairs of participants who were acquaintance (e.g. classmates, friends or colleagues), the rest were strangers.

We believe that these independent variables (especially familiarity) are likely to impact on the length and quality of dialogue, as well as the learning efficiency.

5.3 Statistical Analysis of The Corpus

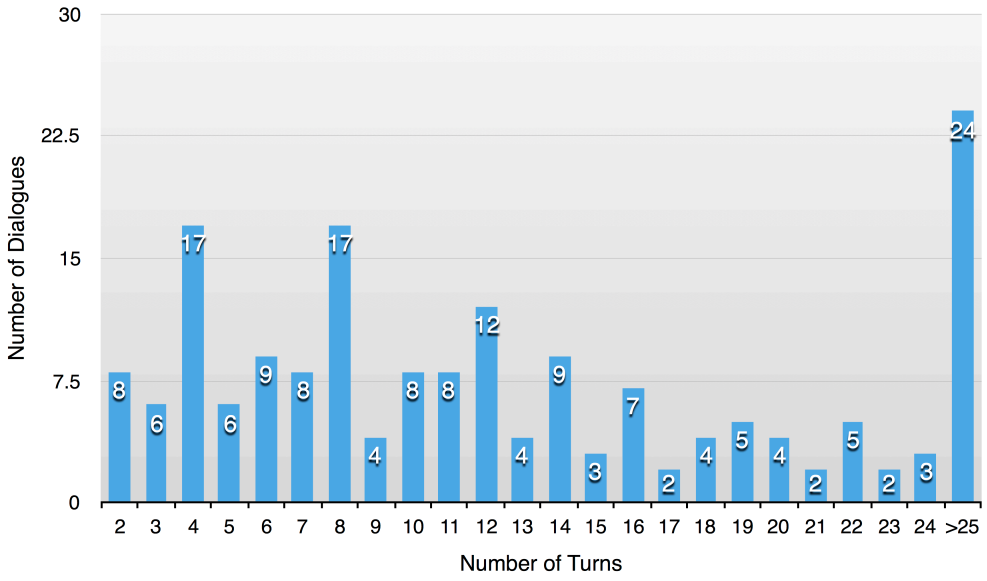


FIGURE 5.2: Dialogue Length Distribution

Using the above procedure, we have collected 177[3] dialogues (each about one visual object) with a total of 2454 turns, where a turn is defined[4] as a sequence of consecutive characters typed by a single participant with a delay of no more than 1 second between the characters. Figure 5.2 shows the distribution of dialogue length (number of turns) in the corpus, where the average number of turns per dialogue is 13.86.

[3]Since there are two pair of participants who didn't complete the task, i.e. only went through 7 and 8 visual objects respectively in the task, instead 180, we only collect 177 dialogue.

[4]Note that the definition of a 'turn' in an incremental system is somewhat arbitrary.

In this section, we statistically analyse this human-human dialogue dataset on two core aspects: dialogue phenomena in the field of incremental language processing (see section 5.3.1) as well as dialogue strategy on the learning task (see section 5.3.2). We briefly discuss the effects of two independent variables on the dialogue collection in section 5.3.3.

5.3.1 Dialogue Phenomena in Incremental Language Processing

As noted above, the DiET Chattool is designed to elicit and record conversations that resemble face-to-face dialogue. In this thesis, we report specifically on a variety of dialogue phenomena that arise from the incremental nature of language processing, as shown below:

- **Overlapping**: where interlocutors speak/type at the same time (the original corpus contains over 800 overlaps), leading to jumbled up text on the DiET interface (see Table. 5.3).

T: this is a ... [[sako bu]]rchak. [[burch]]ak.
L: [[a waka]]ki? [[sorry?]]
T: no, sako burchak,
L: okay, got it.

TABLE 5.3: Dialogue Example of Overlapping

- **Self-Correction**: a kind of correction that is performed incrementally in the same turn by a speaker; this can either be conceptual, or simply repairing a misspelling or mis-pronunciation (see example in Table. 5.4).

T: this is a sako ... no wait ... kasaji wakaki
L: okay, kasaji wakaki.
T: yes, well done.

TABLE 5.4: Dialogue Example of Self-Correction

- **Self-Repetition**: the speaker repeats words, phrases, even sentences, in the same turn (see example in Table. 5.5)

T: this is a kasaji ... kasaji aylana
L: got it, a kasaji aylana.
T: yes, well done.

TABLE 5.5: Dialogue Example of Self-Repetition

- **Continuation (aka Split-Utterance)**: the speaker continues the previous utterance (by herself or the other) where either the second part, or the first part or both are syntactically incomplete (see example in Table. 5.6).

(a) Continuation	(b) Completion
L: what is this? T: a sako wakaki. L: okay. T: sako describes colour and wakaki is the shape. L: great, got it.	T: this is a sako L: wakaki? T: wakaki. wakaki is the shape. L: okay, got it.

TABLE 5.6: Dialogue Example of Continuation

- **Filler**: allows the speaker to further plan his/her utterance while keeping the floor. These can also elicit continuations from the other (Howes et al., 2012). This is performed using tokens such as 'urm', 'err', 'uhh', or '. . . ' (see example in Table. 5.7).

L: en ... is it a sako kasaji? T: no, kasaji describes color not shape. try again? L: uhh, a sako wakaki? T: yes, good job.

TABLE 5.7: Dialogue Example of Filler

As discussed above, although the dialogue phenomena here were produced because of different reasons from spoken language, both of them are presented in the similar format. Hence, for annotating self-corrections, self-repetitions and continuations, we have loosely followed protocols from Colman and Healey (2011), Purver et al. (2009b). Figure 5.3 shows how frequently these dialogue phenomena occur in the BURCHAK Corpus. This figure excludes Overlaps which were much more frequent: 800 in total, which amounts to about 4.5 per dialogue.

Comparing to the other phenomena (e.g. *self-repetition, filler, continuation*), "*self-repair*" is the one that more easily leads to misunderstanding of the tutor's utterance, which may result in worse user experience and even task failures. For example, given the visual-concept learning task in this BURCHAK corpus, the tutor provides a correct description about a specific object but with a self-repair, like "so this is a sako oh no suzuli square.", if the system cannot properly process such phenomena, it might take the word "sako" rather than "suzuli" to update the classifier, which leads to worse classification accuracy and may even ruin the entire knowledge-base (see more details in Chapter 9 which explains what makes such difference between the *"self-correction"* phenomena and others by explaining the incremental processing procedure with the DS-TTR module). In this thesis, we will mainly focus on exploring and evaluating an incremental processing solution on the "*self-correction*" phenomena. We implement an optimised learning agent incorporating the DS-TTR model, which can not only achieve good overall performance through an interactive

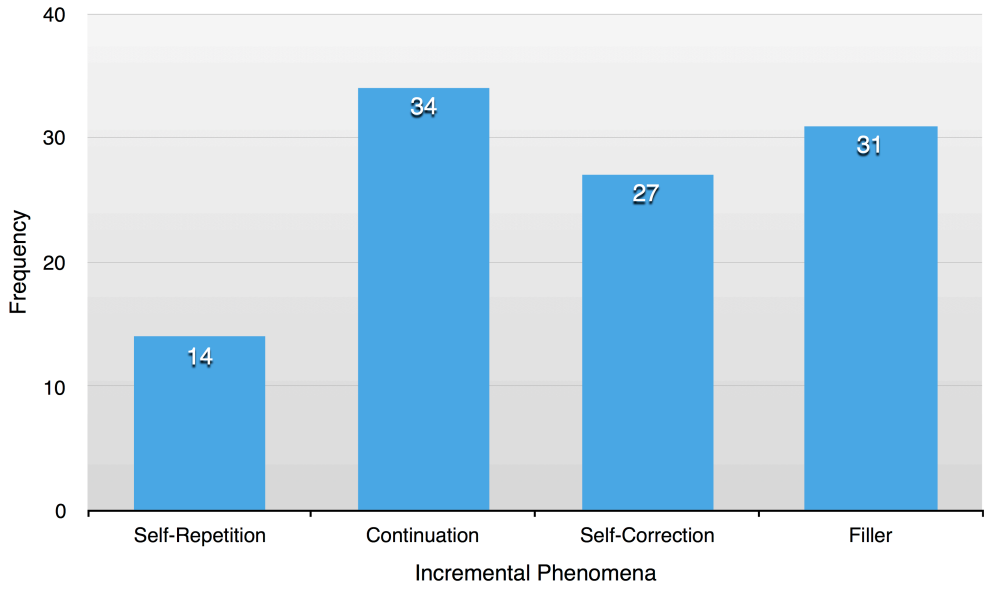


FIGURE 5.3: Frequencies of Dialogue Phenomena (from incremental language processing) in the corpus

life-long learning period, but also process the "*self-correction*" dialogue phenomena within natural, daily conversations with human tutors (see Chapter 9).

5.3.2 Dialogue Strategy

As our domain is based on the task of interactively learning the visual attributes, apart from the incremental dialogue phenomena, we are also interested in how humans can efficiently learn the novel knowledge through conversations (dialogue strategies and capabilities). This section will describe a variety of strategies and capabilities found in this corpus.

5.3.2.1 Learning/Tutoring Strategies

As noted in Section 5.2, each pair of participants needed to learn unseen visual-attribute words by talking through only nine real objects under a tight time constrain – 30 minutes. The number of words correctly mapping to the visual scene was taken into account in deciding the winner in the experiments. Hence, choosing a learning/tutoring strategy may impact the efficiency of remembering as many words as possible in a short period. Both learners and tutors applied these strategies. Here we report a list of behaviours/conditions related to these learning strategies as follows:

Initiative is a basic dialogue strategy that reflects who takes the initiative in a single dialogue. The speaker who takes initiative will be able to positively drive the whole dialogue. When the learner has initiative, it is the one that drives the conversation for one word, by

making a statement about the attributes of the object, asking questions to the tutor for information or confirmation (see Fig. 5.8b), initiates topics etc. On the other hand, when the tutor takes initiative, the learner will wait for the tutor to ask questions to the learner or to make a statement about the attributes of the object (see Fig. 5.8a).

(a) Tutor-Initiative	**(b) Learner-Initiative**
T: let's start with the shape, it is a burchak. L: burchak, okay. T: yes.	L: hmmm, I think this is a wakaki too. T: yes it is.

TABLE 5.8: Dialogue Examples of Initiative in the Corpus

This strategy was used frequently either by the tutor or by the learner across all dialogues, the frequency distribution is shown in Fig. 5.4. It shows that the tutor took around twice as many opportunities to have the initiative as the learner did in dialogues, especially in the beginning of the experiment. This might be because the learner had lower confidence in describing colours or shapes in the unknown language. Most of learners passively received the novel knowledge from the tutor rather than making guesses blindly in the beginning. However, since their confidence give progressively during the learning process with more examples, there were more instances where the learner took the initiative instead.

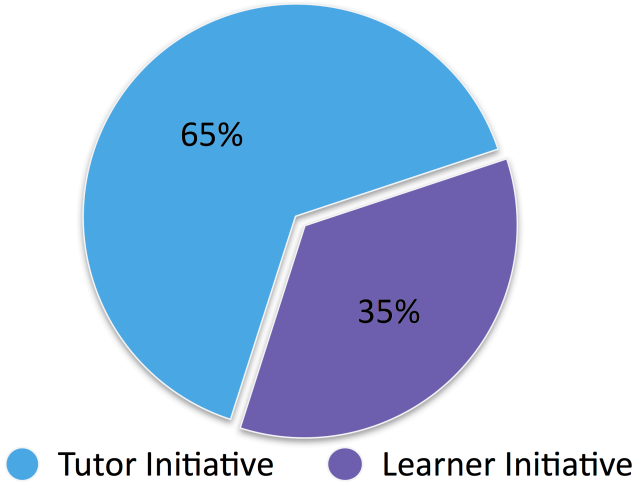


FIGURE 5.4: Initiative Distribution in the Corpus

Context-Dependency is the capability of humans to parse and produce context-dependent expressions, e.g. short answers (see example in Table 5.10a) and incrementally constructed turns (see Table 5.10b). In the learning experiment, using context-dependency allows the participant to save time under a tight time constraint.

Uncertainty Expression, is a basic dialogue strategy that may be triggered while the participant was not highly confident in his/her own answer. In this corpus, the strategy of uncertainty expression frequently occurs on the learner side, especially in the first several

(a) Short Answer	(b) Incremental Turn
T: what is this? L: a sako wakaki? T: yes	T: this is a ... L: sako burchak? T: well done.

TABLE 5.9: Dialogue Example of Context Dependency in the Corpus

conversations on the learning task, when the learner only has ineffectual memory of specific visual attribute words in the made-up language. Hence, the frequency of this strategy gradually decreases with prolonged interaction.

(a) Uncertainty Expression	(b) Certainty Expression
L: is this a sako wakaki? T: yes, well done	L: i know this color, it is sako. T: good job. and shape? L: it is aylana. T: nope, it is burchak. L: okay, sako burchak. got it.

TABLE 5.10: Dialogue Example of Uncertainty/Certainty Expression in the Corpus

Knowledge-Acquisition, also called knowledge demanding, is a learning strategy that is frequently applied by the learner to request further details/information about knowledge – words for either for *new unknown* or *seen but uncertain* attributes. It contributes to a more active learner who can efficiently acquire useful information from the tutor in the learning tasks. In dialogues where knowledge-acquisition is used, the learner is able to request more information by asking extra questions (Table 5.11) show a dialogue example where the knowledge-acquisition strategy was applied in the corpus.

Knowledge Acquisition
T: it is a burchak. L: okay, burchak. and colour? T: the colour is sako. L: okay, a sako burchak. T: yes.

TABLE 5.11: Dialogue Example of Knowledge Acquirement in the Corpus

Knowledge-Review is a new learning-oriented dialogue strategy on tasks, which supports both tutors and learners in reviewing the knowledge learned in the previous dialogue. This strategy is normally applied to the learner after he/she learns several object/attribute words in the collected dialogues. Specifically, the knowledge review strategy can be grouped into 3 methods:

1. ***general review*** – test learned knowledge on all previously learned attribute words by requesting a list of memorized objects (see Table 5.12a).

2. ***specific-attribute review*** – test/remember existing knowledge on specific object/attribute words from the last dialogue by asking WH or polar questions (see Table 5.12b).

3. ***exclusion review*** – test knowledge by excluding incorrect information (see dialogue example in Table 5.12c).

(a) General Review	(b) Specific-attribute Review	(c) Exclusion Review
T: what objects have we seen? L: a sako buchak, a sako wakaki and a kasaji wakaki. T: well done	T: what was the colour of the last object? L: sako? T: no, it was kasaji.	T: what colour is not this object? L: sako T: and? L: kasaji? T: yes, it is called suzuli.

TABLE 5.12: Dialogue Examples of Knowledge-review in the Corpus

Figure 5.5 shows a frequency of three knowledge-review methods in the realistic corpus.

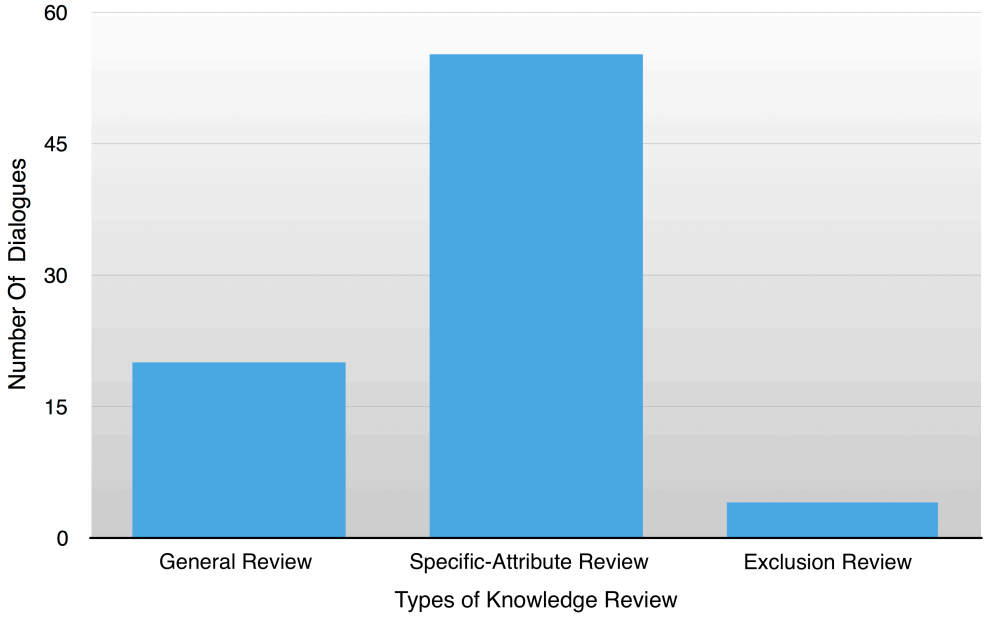


FIGURE 5.5: Dialogue Frequencies of three kinds of Knowledge-review

It is important to highlight that the knowledge-review rarely happens in long-life learning scenario, where the system/robot is required to learn thousands of visual objects and words from the external world.

5.3.3 Effects of Independent Variables

As noted above, the experiment design takes into account two independent variables, *native language* and *familiarity*. We believe that these variables will influence the dialogue

collection. This section briefly discusses their effects on the quality and variations in the corpus.

5.3.3.1 Native Language

Native Language, as an independent variable, is to distinguish whether the participant's first language is English or not. Generally, the native English speakers can better understand the task instructions and also provide higher quality of the uttered sentences (without or with fewer spelling and grammatical mistakes) than non-native speakers did, as explained below:

In terms of the understanding of the task instruction, some of non-native participants has misunderstood the task requirements: they tried to describe the visual attributes using other similar objects or using other languages during the experiment, but which rarely occurs with native participants in our experiment (see examples in Figure 5.13).

(a) Using Similar Objects	**(b) Using Other Languages**
T: sako is the colour of fire. L: okay, sako. T: yes. and burchak is a shape with 4 equal sides, L: burchak	T: the word for the colour is similar to the word for Japanese rice wine. except it ends in o. L: sake? T: yup, but end with an o. L: okay, sako.

TABLE 5.13: Example of Dialogue Snippet with the Misunderstanding of the Task

In terms of the utterance quality, non-native speakers are more likely to have spelling and grammatical mistakes than the native speakers (although with the incremental chat tool, which does not model a delete functionality, is easier to make spelling errors than with the traditional one). An example of a dialogue snippet with a grammatical error is shown in Table 5.14

Dialogue with a Grammatical Error
T: same colour but different shape, it is wakaki. L: okay, a wakaki sako. T: yes, wakaki is the shape and sako is the colour. L: okay.

TABLE 5.14: Example of Dialogue Snippet with a Grammatical Error

5.3.3.2 Familiarity

Familiarity is an independent variable that differentiates participants in pairs who are acquaintances or strangers. The level of familiarity between participants in each pair is likely to affect the length and even the quality of the dialogues in the corpus. In our experiment, participants who are not familiar with each other tend to be more polite to chat with each other using a formal language on the chat tool (see example in Table 5.15a). Compared to the others, they can concentrate more on the task itself. On the other hand, participants, who are classmates/friends, were more relaxed and had more task-unrelated conversations or used more informal language in the experiment. These participants, in contrast with the others, are accustomed to talking with each other using some personalised symbols, such as emoticons, special symbols and even abbreviations in the conversations, most of which are unparsable and unproducible in the further development of a teachable dialogue system (see Table 5.15b).

(a) Between Strangers	**(b) Between Acquaintance**
T: this is a sako burchak. L: ok. T: the sako is a kind of color and burchak is a kind of shape. L: ok I understand.	L: ok this is wakaki suzuloooo. T: no, the color is @suzuli. L: ok, suzuli suzuli :P. T: yes, like the past* pasta L: yes, pasta yummy!!

TABLE 5.15: Example of Dialogue Snippets on the Condition of Familiarity

Compared to an unfamiliar pair of participants who have clean and short dialogues, familiarity is more likely to lead to a longer and more complex conversation in a single dialogue.

We plot a comparison of the dialogue-length between familiar and unfamiliar participants pairs, see Fig. 5.6. It shows that there are about 32 dialogues by the familiar participant pairs that contains over 18 turns, which is much more than that by the unfamiliar ones. Figure 5.6 indicates that familiar participants may lead to more dialogues which are also longer than unfamiliar pairs.

5.3.4 Corpus Processing & Dialogue Capability

As noted in the beginning of this chapter, we collected these realistic dialogues to build an interactively teachable system which can optimise its behaviours to effectively learn unseen visual attributes from humans. In order to achieve this, we will train a simulated tutor based on the collected data. However, since the original corpus was collected at the character level, dialogues in this corpus are messy and contain some unreproducible snippets as described in section 5.3.2.1. Therefore, in this section, we look into how the collected dialogues are

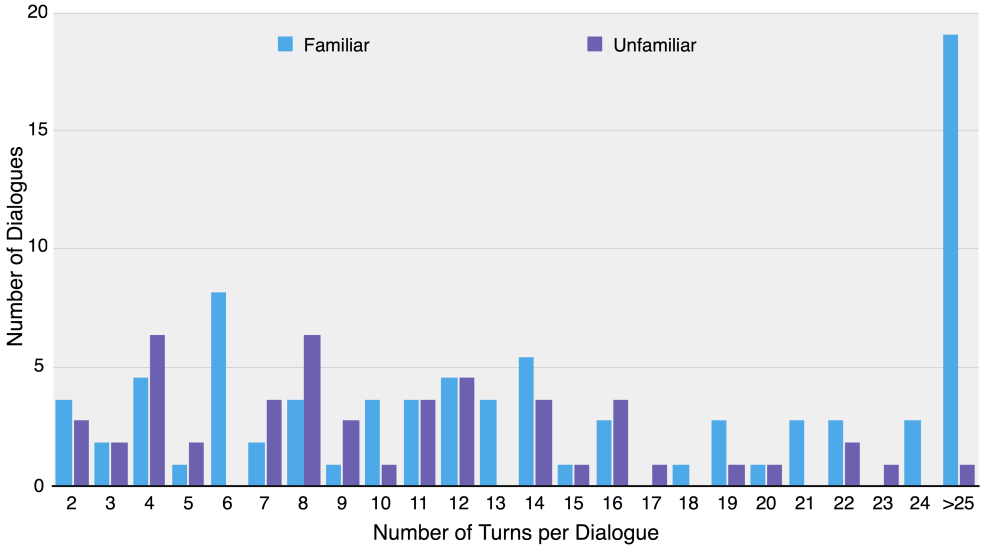


FIGURE 5.6: Dialogue Length Distribution between familiar and unfamiliar participants

processed (i.e. cleaned up and annotated manually), and also what dialogue capabilities and phenomena require further research and development in the thesis.

5.3.4.1 The Data Clean-up Procedure

Following the analysis above, the original corpus contains a list of features which are unusable for building a robust dialogue system in our research. Hence, for the purpose of training the user simulation, and subsequent training of dialogue policy using Reinforcement Learning described in the following chapters, we cleaned the original corpus as follows:

1. we fixed the spelling and grammatical mistakes which were not yet repaired by the participants themselves (see Table 5.13);
2. we removed snippets of conversations where the participants had misunderstood the task (e.g. trying to describe the objects using other similar objects or using other languages) (see Table 5.14);
3. we remove the personalised symbols, like emoticons and special symbols from the dialogues, and replace the abbreviations with full expression (e.g. words or phrases) to make the utterance more comprehensive and useful. (see Table 5.15b)

In addition, we also remove dialogue snippets where the participant performs a general knowledge-review and an exclusion review, as they rarely happen in teachable system which aim at learning a large number of visual attributes/objects from the physical world.

5.3.4.2 Dialogue Capability

Following the cleaned-up version of the BURCHAK corpus, the data set contains a number of capabilities (around 16 capabilities). Some capabilities/actions, such as *inform*, *acknowledgement/rejection*, and *correction*, are defined as basic capabilities that are commonly encountered in standard task/goal-oriented conversations. The others, e.g. the *repetition-request* and the *retry-request*, are related to particular strategies in this learning task.

Here, as our corpus contains a number of strategy-oriented capabilities, which were manually annotated with regards to the conversational behaviour of both the tutor and the learner. We loosely followed some of the existing annotation scheme[5] (e.g. by Buckley and Wolska (2008), Mitchell et al. (2012)) for the tutorial dialogue to design a simple annotation scheme that contains 14 dialogue capabilities found in the BURCHAK corpus (see Table 5.16). We make an use of this scheme to annotate all conversations collected in this experiment, and also to name dialogue actions for training both user simulation and dialogue strategies in the further work.

Dialogue Capability	**Speaker**	**Label**
Listen	Tutor/Learner	Listen()
Inform	Tutor/Leaner	Inform(colour:sako&shape:burchak)
Question_asking	Tutor/Leaner	Ask(colour), Ask(shape), Ask(colour&shape)
Question-answering	Tutor/Leaner	Inform(colour:sako), Polar(shape:burchak)
Acknowledgement	Tutor/Learner	Ack(), Ack(colour)
Rejection	Tutor	Reject(), Reject(shape)
Focus	Tutor	Focus(colour), Focus(shape)
Clarification	Tutor	CLr()
Clarification-request	Learner	CLrRequest()
Help-offer	Tutor	Help()
Help-request	Learner	HelpRequest()
Checking	Tutor	Check()
Repetition-request	Tutor	Repeat()
Retry-request	Tutor	Retry()

TABLE 5.16: List of Dialogue Capabilities and corresponding Annotation labels for the BURCHAK Corpus

These capabilities are explained in detail as follows:

- ***Listen***: is viewed as a special dialogue capability where the participant keeps silence to release the floor to the other speaker. Although both the tutor and the learner are able to call this action, it was more frequently performed by the learner during the

[5]Unfortunately, because of the time limitation, we did not validate the annotation scheme with multiple annotators in advance.This will be addressed as part of the future work.

experiment, especially when the learner did not have knowledge about visual attributes in the beginning of experiment;

- ***Inform***: is a basic capability that makes a statement to describe specific visual attributes (known as slot values) with or without a category clarification, e.g. "it is a sako burchak." or "the color is wakaki, but the shape is burchak.";
- ***Question-asking***: is one of the basic dialogue capabilities, where a participant ask for particular values/visual attributes via either a WH question or a polar question, such as "what colour is this?", "the colour was?" and "it is what?";
- ***Question-answering***: is a capability corresponding to the capability of question-asking, where a participant answers a question about one or more specific values. It can be done using the Inform or asking a polar question, for example,"T: what colour is this? L: wakaki." or "L: is this object a burchak? T: a yes?";
- ***Acknowledgement/Rejection***: a pair of capabilities that process confirmation or negation respectively on a given visual attribute, e.g. "yes, well done." or "no, the shape is right, but the colour is wrong.";
- ***Correction***: is a basic dialogue capability frequently called by the tutor in the experiment to provide the learner correct answers using statements. It normally follows the action of "rejection", e.g. "L: a sako wakaki? T: no, the shape is wrong, it is a burchak.";
- ***Focus***: is defined as a topic-switch capability that keeps the dialogue stay on topic or switches it topic to a particular slot-value, e.g. "now, let's learn the shape." or "it has a similar shape". It was frequently used by participants in the role of the tutor.
- ***Clarification & Clarification-request***: are a pair of capabilities, which allow the speaker to make/request a further explanation on particular slot-values, e.g. "so wakaki is the shape not colour." or "sako describes colour?"; The clarification is mainly performed by the tutor, and the clarification-request by the learner.
- ***Help & Help-request***: a pair of capabilities that allows the speaker to provide or request help from the other participants in the conversation, e.g. "L: it is a sako ... need help. T: burchak." or "L: what is this? L: en... T: need help? L: yes please. T: burchak.".
- ***Checking***: was frequently called by the tutor to make sure whether the learner had understood and learned what he/she was talking about in the previous dialogue turn, for example, "the colour is sako, got it?".

- ***Repetition-request***: is a capability that allows the speaker to ask the other participant to repeat what was said by either participant in previous conversations. It is related to the learning strategy of the knowledge-review (as described in Section 5.3.2.1), in which the tutor attempts to test what the learner has learned in the previous conversation by asking him/her to repeat the specific attribute word over time, e.g. "can you repeat it again?".

- ***Retry-request***: is a capability frequently called upon the tutor to ask the learner to try to describe the visual attribute again when the previous description is incorrect, e.g. "no, it's wrong, try again?". It can be viewed as a second-chance capability frequently performed by the tutor in dialogues, where the learner can be provided another chance to guess correct words for the visual scene instead of being told the correct answer immediately. This might be because the tutor believes that the learner's memory can be efficiently activate if he/she is not directly given the correct labels.

In order to simulate these capabilities properly for training the dialogue system, we annotated the cleaned-up corpus with two types of labels: 1) the "id" for each pair of the learner and the tutor turns, and 2) dialogue actions, slot types (visual attribute categories, e.g. colour and shape) and slot values (specific visual attributes) for each utterance. Table 5.17 shows an example on how the annotation schema is applied to tag each dialogue manually.

Index	Speaker	Utterance	Annotation
1	Learner:	<None >	Listen()
2	Tutor:	it is sako	Inform(colour=sako)
3	Learner:	okay, sako.	Ack()&Repeat()
4	Tutor:	yes. that's the name of the colour. the shape now.	Ack()&Clr(color)&Focus(shape)
5	Learner:	burchak?	Polar(shape=burchak)
6	Tutor:	well done.	Ack()

TABLE 5.17: Example of Annotation Schema in the cleaned-up BURCHAK Corpus

Through annotation and further statistical analysis, some capabilities, such as *Inform*, *question-ask* as well as *acknowledgement/reject*, frequently occurred on both the tutor and the learner sides in the experiment. Fig. 5.7 shows how often each dialogue capability occurs in the data set. The corpus shows huge variations on each capability.

On the other hand, Fig. 5.8 shows the frequencies of these actions by the learner and the tutor individually in each dialogue turn. In contrast with previous work that assumes a single action per turn, here we see multiple actions per turn (see Table 5.18). In terms of the *Learner* behaviour, the learner mostly performs a single action per turn. In contrast, although the majority of the dialogue turns on the tutor side also have a single action, about 22.59% of the dialogue turns perform more than one action.

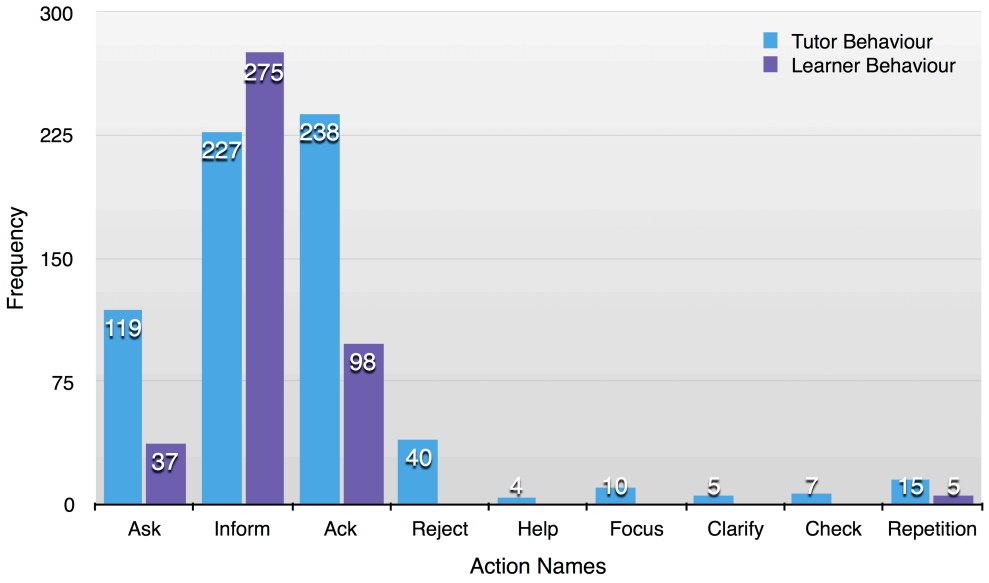


FIGURE 5.7: Frequency of Dialogue Capabilities occur on the tutor and the learner sides in BURCHAK

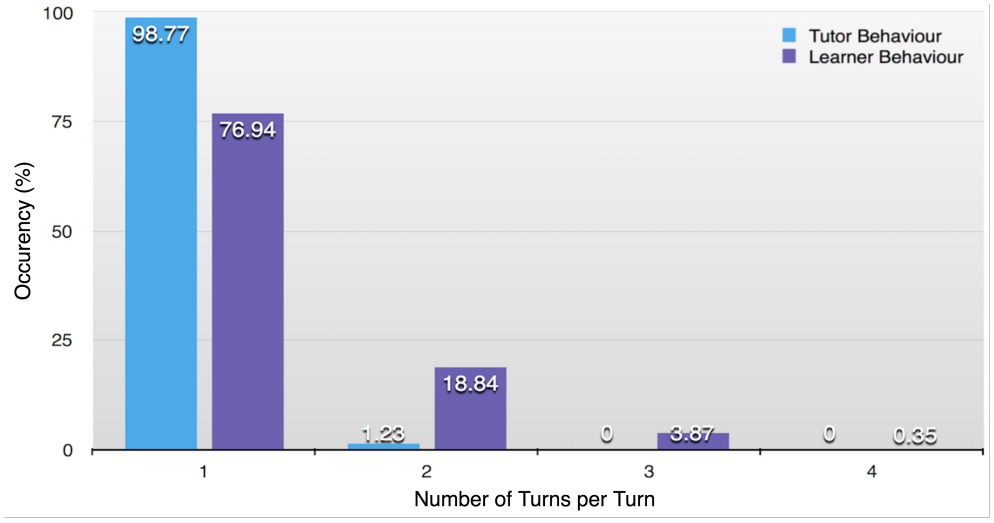


FIGURE 5.8: Distribution Statistics of Multi-Action occurs in BURCHAK (It plots the distribution statistics from the tutor and the learner behaviours separately)

Multi-Action Example	**Annotation**
L: so this shape is wakaki? T: yes, well done. let's move to the colour. So what colour is this?	Polar(shape=wakaki) Ack()&Focus(color)&Ask(color)

TABLE 5.18: Dialogue Example of Multi Dialogue Actions in the corpus

5.4 Discussion: Text Chat versus Spoken Interaction within the Visual Learning Task

In this chapter, we collect a number of natural human-human conversations via an incremental, non-editable, text-based chat tool (DiET), which, different to most standard chat interfaces (e.g. WhatsApp or Facebook Messenger), supports unstructured dialogue turns between humans (see Fig. 5.1). Although such functionality makes our corpus closer to face-to-face conversations, we acknowledge that there are potentially important differences between the collected dialogues and natural spoken conversations. Here we mainly discuss some of the differences that arise in 1) the kinds of dialogue action and strategy that the participants perform; and 2) the linguistic phenomena that the incremental, non-editable, text-based chat tool gives rise to.

In terms of dialogue actions, within a specific conversation domain, the dialogues we have collected can be said to be natural, because the same actions are available to achieve the final goal, whether this is through written or spoken interaction. For example, given the visual-attribute learning task in this thesis, in order to eventually learn how to identify and describe the colour and shape of a certain object, the *Learner* usually performs dialogue actions such as "*informing*" the colour and shape of an object, "*asking*" or "*answering*" questions, "*requesting*" clarification, help and repetition from the fellow tutor; and the tutor also performs actions to provide further feedback or help, for example, "*acknowledgement*", "*rejection*", "*correction*", etc. (see more details in Section 5.3.4.2).

On the other hand, the interface is text-based with specific properties and this affects the *conversational process*, e.g. turn-taking or adjacency patterns. Importantly for us here, this also shows itself in the pattern and frequency of different incremental phenomena that occur in naturalistic dialogue, e.g. *self-repair*, repetitions, *continuation* etc. (see Section 5.3.1 for a list of different phenomena). We explain this in more detail below:

- **Pattern**: Compare to the spoken dialogue corpora (e.g. the BNC (Leech, 1992) and the HCRC Map Task (Anderson et al., 1991) corpora), the dialogue phenomena, such as "self-repair", is presented in the similar format but with a different purpose: most of "self-repair" phenomena in spoken dialogues are usually caused by contextual/conceptual mistakes – providing incorrect information about specific objects or events, but those from the BURCHAK corpus are mostly caused by grammatical or spelling mistakes, for example, "Learner: this is a saki ... no ... sako wakaki.". This might because that, given the incremental, non-editable chat tool, the user does not have enough time to structure his sentence and even words while typing in, which leads to more spelling mistakes, which are never encountered within spoken dialogues.

- **Frequency**: In terms of the frequency of dialogue phenomena (e.g "self-repair", "continuation", and etc.), we also compare our corpus (written) with the BNC (Leech, 1992) and the HCRC Map Task (Anderson et al., 1991) corpora (spoken). The results show that both "self-repair" and "continuation" are preferred within spoken conversations over them in text-chat ones: 1) "self-repair" gives an average of the frequency of more than one event per 2.5 turns within the HCRC Map Task corpora (Purver et al., 2009b), but every 80 turns in the BURCHAK corpus; and 2) Similar to the "self-repair", the phenomena of "continuation" also occurs less frequently through text-based conversation than spoken ones.

In this thesis, we have had to manually modify the spelling-mistake "self-repair" phenomena into the conceptual ones. Since our work here focuses more on how the participant cope with these phenomena rather than the dialogue phenomena themselves, we were willing to accept the aforementioned differences between written and spoken dialogues, which affects neither the ultimate research goal of this thesis, nor the overall performance of the learning agent (see Chapter 9).

5.5 Chapter Summary

This chapter provided a review of existing dialogue corpora that involve realistic human-human conversations on a variety of domains/tasks, however, none of them can be applied/transferred into our domain, suitable for training multi-modal conversational agents that perform the task of *actively learning visual concepts* from a human partner in *natural, spontaneous* dialogue. Therefore, in this chapter, we reported on our work on a new challenge human-human dialogue dataset – BURCHAK – that aims at tracking on the visual attribute learning task, using an incremental dialogue experimental toolkit (DiET (Healey et al., 2003, Mills and Healey, 2017)). We designed a task-oriented experiment for collecting realistic dialogues, where pairs of participants are required to teach/learn the attributes of visual objects (e.g. colour and shape) through conversation. This task is similar to a second-language learner who learns to describe visual attributes in a made-up language, such as "sako" for red and "burchak" for square. Through the experiment, we investigate the impact of several independent variables (e.g. native language and familiarity) on the nature and complexity of conversations: non-native speakers and strangers are more likely to lead to a short and clear dialogue.

Apart from this investigation, the corpus makes three key contributions, as follows:

- To our knowledge, this corpus is the first human-human dialogue corpus that addresses the visual-attribute teaching/learning task through natural, spontaneous dialogue;
- The corpus contains a list of dialogue strategies and capabilities on both the tutor and the learner sides, which may impact the learner's performance on the learning task (see Chapter 7);
- The most challenging aspect of this corpus is that it contains a wide range of dialogue phenomena of incremental language processing, e.g. overlapping, self-repair and self-repeating.

In the next chapter, we will introduce a generic user simulation framework for building a user model that is able to reproduce user behaviours on the interactive learning task. The proposed user simulation is also able to produce the incremental dialogue phenomena described in section 5.3.1. This user model will be applied to train an optimised attribute learning agent in further research.

Chapter 6

Incremental Tutor Simulation

In this chapter, we implement a basic and generic user simulation model using n-gram algorithm, which aims at resembling human-tutor behaviours on teaching novel visual knowledge through dialogue (as observed in the BURCHAK corpus (Chapter 5)). This tutor simulation will be applied to train and evaluate dialogue systems with realistic conversations, for example, an optimised learning agent (see details in Chapter 8). This simulation is generally designed and implemented from two aspects:

1. **Multi-level Simulation**: In the last decades, there has been a surge of interest and significant progress has been made on a variety of simulation approaches, including predicting the next word/full utterance in the user response (Chung, 2004, Schatzmann et al., 2007b), inferring a sequence of user actions in a more abstract level (Ai and Weng, 2008, Asri et al., 2016, Chandramohan et al., 2012, Cuayáhuitl et al., 2005, Eckert et al., 1997, Eshky et al., 2012, Georgila et al., 2005), as well as modelling or producing the user behaviour in semantic representations (Schatzmann et al., 2007a,c). Here, we attempt to build a generic simulation that can, not only produce coherent user responses at the utterance- or action- level, but also to be able to predict the next move incrementally, word-by-word from the conversation history.

2. **Dialogue Phenomena Simulation**: In the previous chapter, the BUCHAK corpus collected a number of natural human-human conversations with a wide range of dialogue phenomena that are frequently discussed in Incremental Dialogue Processing, such as self-correction, -repetition, filler, pause and etc.. These phenomena appear as "noise" or "performance errors" in the dialogue context that may lead to interactional and semantical consequences – strongly and directly impacting 1) how conversation partners coordinate their moves and the emergent semantic context of their conversations, and 2) how a dialogue agent may adapt to their partner over time. Hence,

our model should be able to mimic natural, human-like conversations by *inserting* or *predicting* those phenomena from the BURCHAK corpus.

In this chapter, we firstly reviews several existing approaches for building a conversational user simulation (see Section 6.1). In Section 6.2, we then explained how out model is built for resembling both human behaviour and dialogue phenomena in human-human conversations given the learning task. It is followed by an experiment to evaluate the performance of the proposed user simulation on two different dialogue corpora (the BURCHAK corpus (Yu et al., 2017b) and the Facebook bAbi corpus (Weston et al., 2015)) on different levels of abstraction (Section 6.3 and Section 10.1).

6.1 Techniques for User Simulation

In this section, we review a set of techniques for building a user simulation to learn robust dialogue systems, which is one of its fundamental tasks. In order to achieve this, the user simulation is required to mimic as natural human behaviours as possible in the dialogue training process. In the past decades, a number of approaches were introduced to support the implementation of the user simulation. These approaches can generally be categorised based on the level of abstraction at which the dialogue is modelled: 1) the action-level has become the most popular user model that predicts the next possible user dialogue action according to the dialogue history and the user/task goal (Ai and Weng, 2008, Asri et al., 2016, Chandramohan et al., 2012, Eckert et al., 1997, Eshky et al., 2012, Georgila et al., 2005, Levin et al., 2000); 2) on the word/utterance-level, instead of dialogue action, the user simulation can also predict the full user utterances or a sequence of words given specific information (Chung, 2004, Schatzmann et al., 2007b); and 3) on the semantic-level, the whole dialogue can be modelled as a sequence of user behaviours in the semantic representation (Schatzmann et al., 2007a,c).

6.1.1 User Simulation on the Action Level

We firstly describe the most popular approaches that build the user simulation on the dialogue action level. Instead of producing full human utterances, these simulated users predict the dialogue action of the user's next move, which can efficiently satisfy the continuation of interactions between human users and systems.

The action-based user simulation was firstly suggested by Eckert et al. (1997), who introduced a bi-gram model condition user simulation that predicts the next user action (a_u) given a certain previous system action (a_s), as below.

$$P = P(a_u|a_s) \tag{6.1}$$

The proposed bi-gram model is a probabilistic model that is fully domain-independent. Since it does not contains enough constrains on the simulated user, the predicted user responses can be appropriate for the previous system action, but not for the wider dialogue context. Although Eckert et al. (1997) emphasised that the user model is likely to be extended to an n-gram user simulation, it is hardly applied to train an n-gram model with $n > 2$, because of the data sparsity. However, the prediction of the bi-gram model (that the next user action upon previous system actions) seems oversimplified, because the user actions is likely to violate the logical constraints and also the user may continuously change his/her goals or repeating request in real-time long conversations.

Eshky et al. (2012) considered a dialogue as a sequence of turns, with multiple utterances on each. Each utterance is tagged with an action, a slot as well as a slot-value (as exampled in Table 6.1). The model does not constrain the number of contiguous system or user utterances. They proposed a topic-goal model for the goal-oriented user simulation that considered all values for specific slot as a count vector (Eshky et al., 2012). It then was taken as samples from a topic model – Mixture-of-Multinomial model – the possible topic is viewed as a set of samples when each dialogue rather than each value is uttered. The model computes probabilities over act-slot pairs via a "bi-gram-based Act model" (see Eq. 6.2).

Speech	Semantic Representation
M: Hello, How Can I help?	M: GREETING M: META_REQUEST_INFO
U: A trip from New York City to Osaka, please.	U: PROVIDE *orig_city* **New York City** U: PROVIDE *dest_city* **Salt Lake City**
M: Leaving from New York City to Salt Lake City. What day would you like to travel?	M: IMPLICIT_CONFIRM *orig_dest_city* M: REQUEST *depart_date*
U: No, no. Leaving from New York to Osaka in Japan.	U: NO_ANSWER *null* **no** U: PROVIDE *orig_city* **New York** U: PROVIDE *dest_city* **Osaka Japan**
M: Leaving from New York to Osaka Japan, correct?	M: EXPLICIT_CONFIRM *orig_city* M: EXPLICIT_CONFIRM *dest_city*
U: Yes.	U: YES_ANSWER *null* **yes**

TABLE 6.1: An Example of a Dialogue in Speech and its Semantic Equivalent (Eshky et al., 2012)

$$p(u|m) = \prod_s p(Z_s) \cdot \prod_i p(a_i, s_i|m_{i-1})p(v_i|Z_i) \tag{6.2}$$

In the practical implementation, "some slots will not always have corresponding values, or will be slots whose values are not appropriate to model in the above way". Hence, in order to cope with this problem, Eshky et al. (2012) applied: 1) a separate, standard bi-gram model to handle utterances which are not defined in such slots, and 2) the topic-goal model for appropriate utterances.

Asri et al. (2016) proposed a data-driven user simulation using a Sequence-to-Sequence model (see architecture in Fig 6.1). This simulation aims at inferring the possible dialogue agenda, a sequence of dialogue actions, based on previous dialogue context (c). This model consists of an encoder Recurrent Neural Network (RNN) and a decoder RNN, both of which are based on a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997).

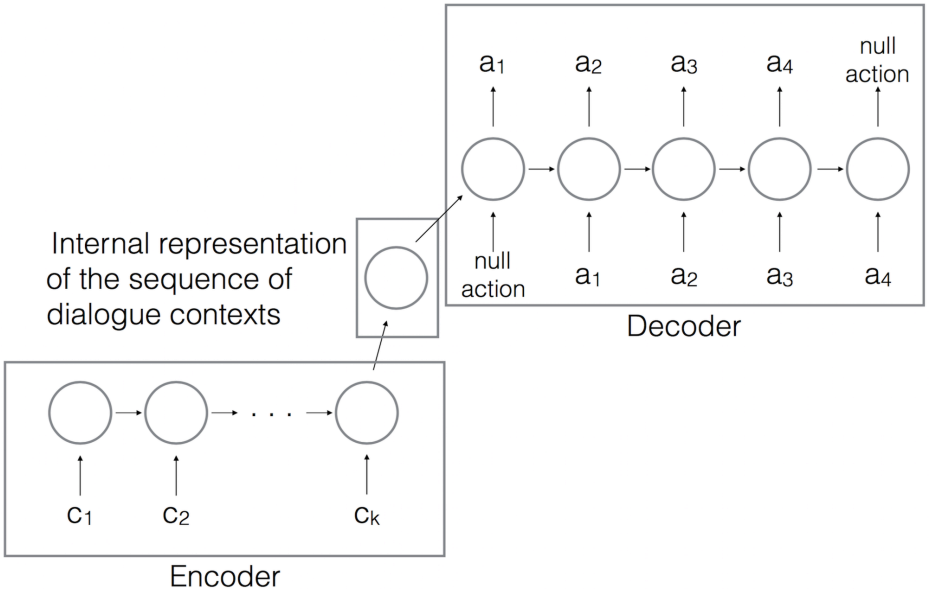


FIGURE 6.1: Architecture of the Sequence-to-Sequence User Simulation Model (Asri et al., 2016)

More specifically, before the model, each dialogue is given a user goal $G = (C, R)$, where C is a set of constrains to inform and R is a set of requests to ask in the conversation. The encoder-decoder model will take the whole dialogue history as input and a sequence of dialogue actions. In terms of the encoder model, it takes a sequence of dialogue context $(c_1, c_2, c_3, \ldots, c_k)$, where a context c_k at turn k is defined by 4 main components, as below:

$$c_k = a_{s,k} \odot inconsist_k \odot const_k \odot req_k, \tag{6.3}$$

where $a_{s,k}$ is a vector of the most recent system action, $inconsist_k$ represents the the inconsistency between the latest information provided by the system and the user goal, $const_k$ represents the binary constraints status (1 for informed, otherwise 0), and req_k for the binary request status (1 for informed, otherwise 0).

In terms of the decoder, each sequence output by the simulator is a sequence of dialogue acts, e.g., *(inform, request)*. Asri et al. (2016) then map these dialogue acts to user slot-value actions and responses by "looking at the current user goal and uniformly drawing among the constraints left to inform and the requests left to ask". However, as this simulation model deploy a deep neural network method that usually requires a huge amount of training data, it is not suitable for our task with the BURCHAK corpus (only 177 dialogues).

Georgila et al. (2005) built a user simulation "with an Information State Update" (ISU) framework, which takes into account the whole dialogue history. It considers a dialogue"as a sequence of pairs of speech acts and tasks" into a dialogue state. The model predicts the next user action based on features of given the current dialogue state (i.e. the $n-1$ most recent $< speechact, task >$ pairs in the history). Similar to our simulation framework, this simulation selects user actions using an n-gram algorithm that maps each state to a specific feature vector. Georgila et al. (2005) compared this method with a baseline which applies a linear feature combination to predict user actions. Through experiments, both the linear feature combination model and the best n-grams (5-gram and 4-gram) are able to produce similar results.

6.1.2 User Simulation on other Levels

We also reviews a set of other simulation models and relevant approaches, which resemble user behaviour at different levels instead of at the dialogue action level, for instance, Ai and Weng (2008), Schatzmann et al. (2007a,b,c).

Schatzmann et al. (2007a) introduced an agenda-based simulation that formalises dialogue at the semantic level as a sequential state transactions and dialogue actions. At any time t, the user is in a state S, transitions into the intermediate State S' by taking action a_u, and then transitions into the next State S'' when the system takes action a_m, where the cycle starts again.

$$S \rightarrow a_u \rightarrow S' \rightarrow a_m \rightarrow S'' \tag{6.4}$$

This model formalises a user state into two key factors, e.g. an agenda (A) and a goal (G). Each goal contains a list of constrains (C) as well as requests (R). Each user agenda consists of several user dialogue actions that are likely to be applied to retrieve information specified in the goal. This model assumes a probabilistic distribution over the user goal, which is either manually set without data Schatzmann et al. (2007c) or introduced from

the realistic data Schatzmann et al. (2007a). It selects an appropriate user action based on this probabilistic distribution, and recalculate the probability of the next state based on the updated user agenda A and also the user goal G.

Ai and Weng (2008) also built a similar user simulation, similar to Schatzmann et al. (2007a)'s work, on the domain of restaurant, which aims at finding an expected restaurant based on some latent constrains specified in the final goal. Whilst, in contrast to Schatzmann et al. (2007a)'s model, they introduced their model at the word level instead of semantic level. The model produces a sequence of words by instantiating the current action using the pre-defined templates. The model applies two error generator with a particular rate (e.g. a random lexical error generator (15%) and a semantic error generator (11%)) to simulate a spoken language understanding performance.

6.1.3 User Simulation on Multi-Level

On the other hand, there are also some user simulations built on multiple levels. For instance, Jung et al. (2009) integrated different data-driven approaches on action- and word -levels to build a novel user simulation. The user-action simulation is to generate user-action patterns, and then a two-phase data-driven domain-specific user-utterance simulation is proposed to produce a set of structured utterances with sequences of words given an action and select the best one using the BLEU score. In the later work Jung et al. (2011), they instead introduced a data-driven user action simulation in Markov Logic framework. It integrates the user knowledge (e.g. cooperative, corrective and self-directing) to generate user-action patterns for the corresponding user-type.

6.1.4 Summary

In this section, We looked through a list of methods for building a user simulation that mimics the human behaviour in realistic conversations for training a robust dialogue system. They concentrated more on well-structured turn-taking dialogues or goal-completion for specific tasks. Those user models are concerned with a wide variations of user behaviours in human-human conversations. However, they rarely takes into account replicating specific, more complicate dialogue phenomena, e.g. overlapping, self-correction, self-repetition, and continuation, which have attracted our attentions in this project (see Chapter 5). Similar to Jung et al. (2009)'s work, in this chapter, we propose a generic user simulation framework on multiple levels, e.g. utterance-, action- and word-levels. But what differ from the other frameworks is that the model should be able to reproduce specific incremental dialogue phenomena in the training process, as described in the following sections.

6.2 Implementation of The N-gram User Simulation

This section explains a simple n-gram user simulation framework that aims at resembling human tutor behaviours observed from the BUCHAK corpus. This model will be used to train and test data-driven dialogue systems, e.g. interactive learning agents in the following chapters. Here, we will mainly explain the model implementation on two aspects: 1) what algorithms are deployed to implement and 2) how the dialogue phenomena will be simulated on utterance and action levels.

6.2.1 N-gram Method

Following the previous work from Georgila et al. (2005), we implement the user model via a simple n-gram algorithm, where the probability ($P(t|w_1,..,w_n,c_1,..,c_m)$) of an item t is predicted based on a sequence of the most recent words $(w_1,\ldots,w_n)$ presented by the system/machine and additional dialogue context status C:

$$P(t|w_1,..,w_n,c_1,..,c_m) = \frac{freq(t,w_1,..,w_n,c_1,..,c_m)}{freq(w_1,..,w_n,c_1,..,c_m)} \tag{6.5}$$

where $c_1,..,c_m \in C$ represent additional status for specific user/task goals (e.g. goal completion as well as previous dialogue context). For example, given the visual-attribute learning conversations from the BURCHAK corpus, context status can be defined by 1) what attribute category (colour or shape) the previous dialogue turn focused on *preContext*, 2) whether the name of colour attribute presented by system is correct or not C_{state}, and 3) whether the name of shape attribute presented by system is correct or not S_{state} (see more details about how these components are configured in Section 6.3).

The probability distribution in Equation 6.5 is induced from the corpus using Maximum Likelihood Estimation, where we count how many times each t occurs with any specific combination of the conditions $(w_1,\ldots,w_n,c_1,\ldots,c_m)$ and divide this by the total number of times t occurs (see Eq 6.5). The simulation will create a N-gram dictionary that contains a set of i-gram maps $0 <= i <= N$.

Smoothing Method In order to reduce mismatch risk, the model is able to back-off to smaller n-grams when it cannot find any n-grams matched to the current word sequence and conditions. In order to constrain the searching space, we applied the nearest-neighbour algorithm to search for the n-gram matches for the unseen system behaviour or conditions by calculating the Hamming distance of each pair of n-grams.

Simulation Output The n-gram user simulation is generic, as it is designed to handle the item prediction on multiple levels, on which the predicted item, t, can be assigned either to (1) a full user utterance (U_t) on the utterance level, (2) a combined sequence of dialogue action (Dat_t); or (3) the next word/lexical token (see Fig. 6.2). The tutor simulation on the word-level is trained to predict fully incrementally on a word-by-word basis. It will predict the next single word based on a sequence of words from the previous system utterance and words the current speaker previously generated.

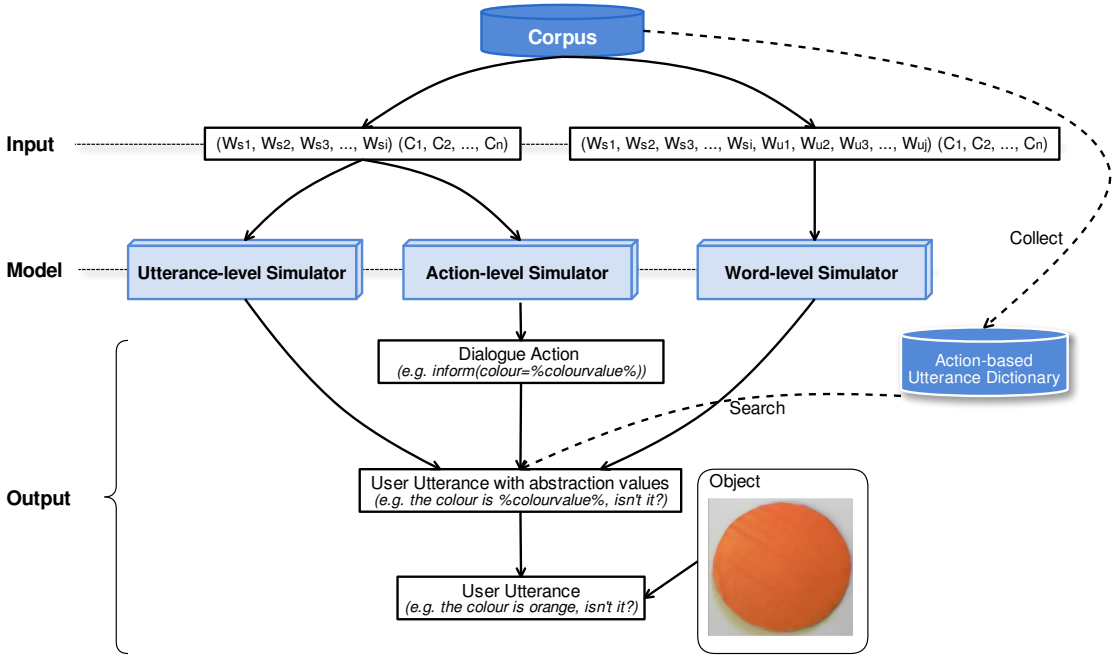


FIGURE 6.2: Illustration of the N-gram Simulation Model (w_{si} represents the i-th most recent word presented by the system, w_{uj} represents the j-th most recent word generated by the user simulation, and C_n represents dialogue context status)

Different with the utterance and word levels, in terms of the action-level simulation, each output by the n-gram simulator is one or multiple slot-value actions, for example, *ask(colour)*, *accept()&polar(shape = %shapevalue%)*, where all colour and shape values are presented on an abstract level, such as, %colourvalue% and %shapevalue% (note that these abstract values are also used on both utterance- and word-level simulation.) For example, in the visual-attribute learning conversation from the BURCHAK corpus, the simulator outputs an action *polar(colour=%colourvalue%)*. We will search for one of possible utterances collected from the corpus[1], "the colour is %colourvalue%, isn't it?". At the end of the simulation process, all abstractions of colour/shape attribute values will be replaced with the

[1]In the beginning of the simulation learning process, the model passes through all dialogues within the corpus, and then creates an action-based utterance dictionary that collects and groups all existing utterances based on their actions (e.g. openask(), ask(colour), ask(shape) and etc.) on the user and system sides separately. When the simulation predicts the specific action for the user, it will search for all utterances matched to the action and randomly choose one as output.

real attribute words of a particular object. for instance, given an object "orange circle", the previous utterance will be updated to "the colour is orange, isn't it?".

6.2.2 Simulating Incremental Phenomena

In contrast to the turn-taking capabilities of traditional dialogue systems where either the system or the user must wait for the speaker to release the floor, current systems require a more natural and human-like ability to incrementally process the speaker's request without waiting for the end of the turn, much like in human-human interaction.

Following the analysis of the BURCHAK corpus, we have dialogue data that contains dialogue phenomena (as described in Section 5.2), which we intend to use to improve our user simulation framework to resemble human-human conversations in this section.

Here, we propose the simulation two approaches to the incremental phenomena: 1) similar to the sequence-to-sequence model, a **word-level n-gram model** can predict the next word (w_t) instead of the next utterance or actions, which makes it easier to simulate the dialogue phenomena, and 2) external **incremental phenomena generation** that adds the incremental features to the utterance in post production. The former method has been explained in the above section – the incremental phenomena will be generated word by word based on the probabilistic distributions of these phenomena in the reference dialogue corpus. Hence, here we look into how the incremental phenomena generation is implemented and applied in the simulation.

Dialogue Phenomena Generator The generator is a randomized simulator that is able to randomly create one type of dialogue phenomena (currently chosen from three possible phenomena:*self-correction*, *self-repetition* and *fillers*) on specific words/phrases based on a occurrence probability that is manually defined. Figure 6.3 illustrates the process of generating those phenomena in the user simulation. The generator firstly decomposes the utterance output by the n-gram model into a sequence of tokens $(w_1, \ldots, w_n)$, and then randomly determines which specific token (w_i) the chosen phenomena will be applied. The selected token w_i will be replaced by the created incremental phrase. Finally, the generator will compose the sequence of words into a full utterance as the outcome of the user simulation. So far this generator is mainly applied to the user simulator on the action level, because the dialogue action simulator produces the clean utterances without the incremental features.

Te main difference between the word-level simulation and the external incremental generation is that, the word-level simulator will process the incremental phenomena immediately after the specific word being produced in the real time, but the incremental generation can

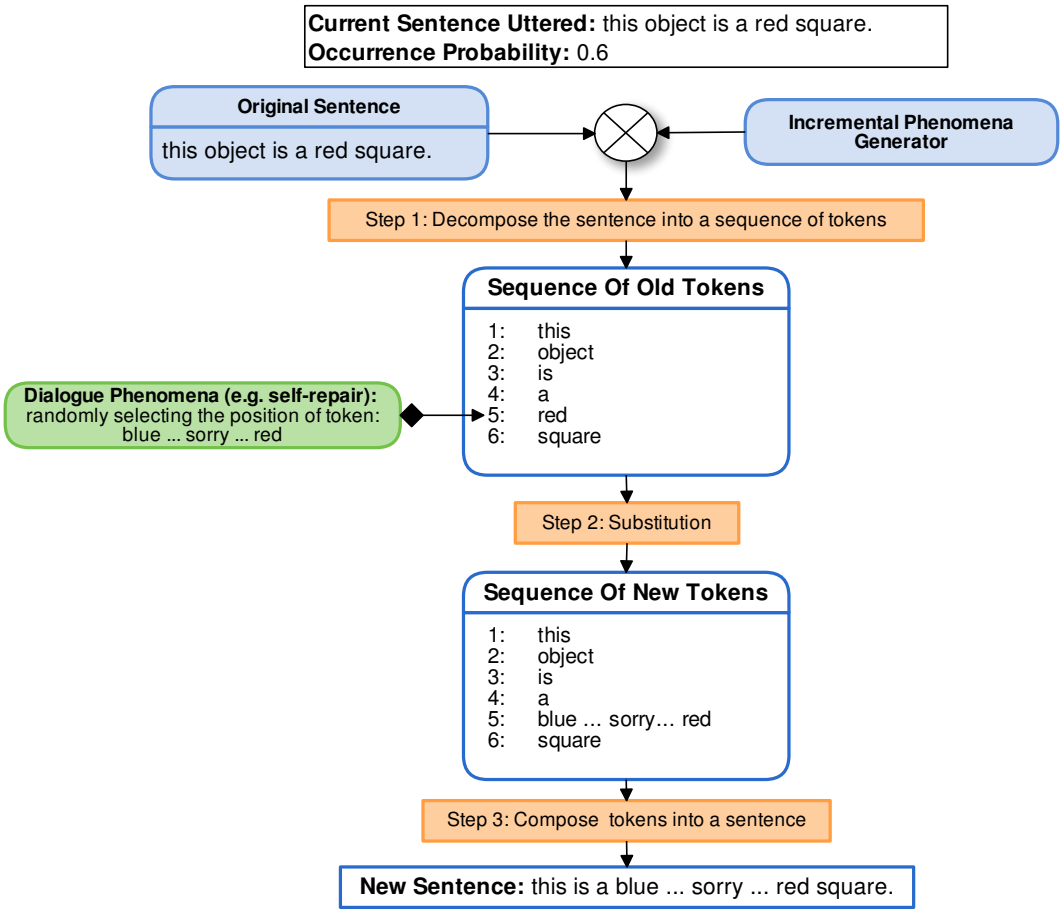


FIGURE 6.3: Illustration of simulating the Dialogue Phenomena from BURCHAK corpus

only add the phenomena after generating the full utterance. In addition, the word-level method is able to produce more kinds of incremental phenomena than the generator.

6.3 Evaluating the User Simulation

In this thesis, we evaluate the user simulation based on turn-level evaluation metrics, where evaluation is done on a turn-by-turn basis. We investigate the performance of the user model at three levels: the utterance, dialogue action and word levels.

6.3.1 Evaluation Metrics of the User Simulation

User simulations were initially introduced to reduce human involvement in the automatic learning and testing process. They are expected to resemble human behaviours as much as

possible, especially when applied to optimise the dialogue strategy, where the user simulation may directly influence on the quality of the learned dialogue policy (see Schatzmann et al. (2007a)). Therefore, ensuring good performance of the user simulators plays an indispensable role in designing a robust user simulation for the dialogue systems. Although there has not yet been a commonly accepted list of evaluation metrics for the dialogue user simulation, some automatic procedures have been introduced to assess their performance. Keizer et al. (2012) suggested that these measures can be classified into two groups, including 1) turn-level evaluation measures (see Section 6.3.1.1) and 2) dialogue-level measures (see Section 6.3.1.2). Although both evaluate the performance of the user simulation based on the quality of synthetic dialogues produced by the user simulator, in contrast with the dialogue-level measures, turn-level evaluation takes into account the local rather than global consistency of the produced dialogues.

6.3.1.1 Turn-level Evaluation Metrics

This section presents the turn-level evaluation metrics, which are the most common methods of evaluation of simulation performance at the level of dialogue transitions, for instance, the prediction capability of user simulations – correlations between the generated dialogues and the corpus dialogues in terms of the action/utterance/word frequency. Here, we introduce several commonly used measures which we will apply to evaluate our framework, as follows:

Accuracy Accuracy (Acc) is a standard method to measures the proportion of correctness, i.e the moves (e.g. an utterance or dialogue acts ($DAts$)) predicted by the simulator can be exactly same as the ones observed from the data, given a particular set of conditions $(w_1, .., w_n, c_1, .., c_m)$. To calculate this, all existing combinations in the data of the values of these variables are used to generate a prediction. If the predicted action or utterance occurs in the data for these given conditions, we consider the prediction correct.

The standard accuracy algorithm requests exact matches, in which a prediction is only considered correct when the user simulation produces an exactly matching sequence of words, phrases and even sentences as they occurred in the corpus given those conditions. However, in realistic, natural conversations (for instance BURCHAK), dialogues may encounter a matching or similar dialogue-action but with different expressions (as these are huge variations on a single action). For example, both "what colour is this object?" and "do you know this colour?" may trigger the same dialogue action "asking colour ($Ask(color)$)" in the BURCHAK corpus.

Since the user model, implemented using an n-gram approach, produces the next user response upon the distributed probabilities, it is more likely that the simulator will generate

a response (especially on the utterance and the word levels), what is slightly different from but shares the semantic meaning with the actual data.

Hence, for coping with this, we introduce another accuracy measurement on semantic level – *Semantic Accuracy* (Acc_{sem}) – which compares the semantic meaning of the response prediction by the user simulation to that of the corpus example. We parse predicted and actual responses into dialogue actions and then compare whether each pair of responses are exactly matched or not[2].

Kullback-Leibler (KL) divergence and dissimilarity Kullback-Leibler (KL) divergence ($D_{kl}(P \parallel Q)$) (Kullback and Leibler, 1951) was introduced to assess the dissimilarity between two probability distributions (see Eq.6.6).

$$D_{kl}(P \parallel Q) = \sum_{i=1}^{M} p_i \log(\frac{p_i}{q_i}) \tag{6.6}$$

KL divergence, also called cross-entropy, was applied in the first place to measure the user simulation in Cuayáhuitl et al. (2005), where P represents the actual probability distributions of user utterance/actions in the realistic dialogues, and Q represents the predicted distribution in the generated dialogues from the user simulation. Since the KL divergence is not symmetric ($D_{kl}(P \parallel Q) \gtreqless D_{kl}(Q \parallel P)$), it cannot be applied to measure the distance between two distributions. A dissimilarity metric ($DS(P \parallel Q)$) is defined below:

$$DS(P \parallel Q) = \frac{D_{kl}(P \parallel Q) + D_{kl}(Q \parallel P)}{2} \tag{6.7}$$

Although KL divergence better measures the similarity between the turn-level dialogue prediction of the user simulation and the real user behaviours than the precision and recall metrics, Keizer et al. (2012) pointed out that "as KL divergence is an unbounded metric, it cannot be applied directly for ranking user simulations".

6.3.1.2 Dialogue-level Evaluation Metrics

This section presents the dialogue-level evaluation metrics. In contrast with the turn-level measures, the dialogue-level metrics measure the performance of user simulations based on features of the entire synthetic dialogues.

[2]In the other words, the Acc_{sem} may rely on the performance of the dialogue action parser applied in the experiment, so thst we implement a simple dialogue act tagging model to support this method for both BURCHAK and Facebook bAbi corpus.

Perplexity Perplexity is widely used as a measure in information theory and was first introduced as an evaluation metric for user simulations in (Georgila et al., 2006). It was proposed to compare probabilistic predictive models, e.g. language models in NLP. The perplexity is defined below:

$$PP = 2^{\frac{1}{N}\sum_{i=0}^{N}\log_2 P_m(X_i)} \tag{6.8}$$

where $P_m(X_i)$ represents the probability of x_i given a predictive model and x_i represents a sample from test set that contains N samples. The higher probability the model gives in the test samples, the lower the perplexity. Hence, the lower perplexity the model achieves, the better the model is. With regards to the user simulation, it will predict a sequence of dialogue acts, utterances and even words, so that the $P_m(X_i)$ represents the probability of a sequence of acts/utterances/words given a user simulation.

Apart from these evaluation metrics, some measures, such as Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) and Simulated User Pragmatic Error Rate (SUPER) (Rieser, 2008, Rieser and Lemon, 2006), are also proposed to measure the performance of the user simulation. However, we will not consider these metrics in this thesis, because they are both highly correlated with the human involvement, i.e. human judgement/rating.

6.3.2 Dialogue Corpus Setup

In order to evaluate the performance of the proposed user simulation framework and prove that this model can be used not only for the BURCHAK corpus itself, we train user simulations on two different dialogue corpora: the BURCHAK (Yu et al., 2017b) and the Facebook bAbi dataset (Weston et al., 2015). We describe the datasets and how they are applied for building the user simulation as follows:

6.3.2.1 BURCHAK

As we described in Chapter 5, BURCHAK contains 177 human-human dialogues on the domain of learning colour and shape attributes. We train user simulations based on the cleaned up corpus (see Section 4.2). It simulates the tutor's behaviour on the learning task. For this specific task – interactively learning visual attributes – the additional dialogue conditions (C) are assigned to the states of both colour (C_{state}) and shape (S_{state}) labels, as well as the previous dialogue context ($preContxt$) related to the visual attributes. More details are shown in Table 6.4.

Condition	Values	Description
C_{state}	Y(es), N(o), U(nknown)	represents whether the color attribute is identified correctly
S_{state}	Y(es), N(o), U(nknown)	represents whether the shape attribute is identified correctly
preContet	U(nknown), C(olor), S(hape), B(oth)	represents which attribute (colour or shape) is currently under discussion.

TABLE 6.2: User Simulation Conditions on the BURCHAK corpus

6.3.2.2 Facebook bAbi Dataset

The bAbi (Weston et al., 2015) is a synthetic human-machine dialogue corpus by Facebook AI Research for the goal of automatic text understanding and reasoning. It is composed of a set of task-oriented dialogues in the domain of restaurants and hotels. For each task, the bAbi dataset contains 1000 dialogues of each training, development and testing respectively (Weston et al., 2015). A dialogue example on the Task 1 – booking a restaurant table is shown in Table 6.3. We build the user simulation to simulate the customer behaviours on the booking task, where $<SILENCE>$, as a special token, represents the missing user utterance, and *api_call* represents a special action for calling APIs for searching restaurant options.

Dialogue Example
U(ser): hello
S(ystem): hello what can i help you with today
U: can you book a table for six people with french food
S: i'm on it
U: $<SILENCE>$
S: where should it be
U: in bombay
S: which price range are looking for
U: i am looking for a cheap restaurant
S: ok let me look into some options for you
U: $<SILENCE>$
S: *api_call*: french bombay six cheap

TABLE 6.3: Dialogue Example in the bAbi Corpus (Weston et al., 2015)

Here we use the Task 1 of the bAbi dialogues to train and test our user simulations. For the task of booking resataurant, the additional dialogue conditions (C) are assigned to several slots, including food type ($Food_{type}$), location ($Location$), number of guests (Num_{guest}) as well as price range ($Price$), as follows:

6.4 Results & Discussion

This section presents the evaluation results of the simulated user model on both BURCHAK and Facebook bAbi corpra. It also attempts to incite a further discussion about the potential

Condition	Values	Description
$Greeting$	Y(es), N(o)	represents whether system greets users
$Food_{type}$	Y(es), N(o)	represents whether food type has been talked
$Location$	Y(es), N(o)	represents whether restaurant location has been talked
Num_{guest}	Y(es), N(o)	represents whether the number of guests has been talked
$Price$	Y(es), N(o)	represents whether whether food price range (e.g. cheap, moderate, expensive) has been talked

TABLE 6.4: User Simulation Conditions on the bAbi corpus (Weston et al., 2015)

effects of user simulation approaches on building dialogue management module.

6.4.1 Results

Table 6.5 presents results on the validation set of the BURCHAK (Yu et al., 2017b) and the Facebook bAbi (Weston et al., 2015) data sets separately. The first observation is that, as expected, the action-based user model achieves good performance on both corpora, because it is more abstract and therefore less sparse. It also produces more variation in the resulting utterances. In addition, both utterance-based and lexical-based user simulations have impressively shown comparable performance to the action-based one for both corpora while comparing them on the semantic level. This means that they can not only produce user responses more naturally, but also can precisely express user actions occurs in the realistic conversations.

Finally, the simulation for the bAbi corpus outperforms that for BURCHAK corpus comparatively, because the bAbi data set, as a collection of synthetic dialogues, contains less variations on all levels (utterance, dialogue action and sequence of words).

	BURCHAK			**Facebook bAbi**		
	Utt-level	**DAts-level**	**Words-level**	**Utt-level**	**DAts-level**	**Words-level**
Acc_{sem}	0.6045	0.6026	0.6070	0.8219	0.8209	0.8182
KLD	0.8869	0.9303	0.4688	0.5027	0.5027	0.1052
$Perplexity$	1.1306	1.1280	1.1140	1.0340	1.0340	1.1331

TABLE 6.5: Evaluation Results

Table 6.6 shows two examples of dialogue produced by the user simulation on the dialogue-act level. These examples present coherent dialogues with incremental dialogue phenomena on the attribute learning task and the restaurant booking task respectively. Although the bAbi corpora only contains clean synthetic dialogues, the user simulation can also generate some incremental phenomena in the output.

Dialogue Example (a)	Dialogue Example (b)
T: what is this object called?	U: hello
L: a sako ... emm ...	S: hello can i help you
T: need help?	U: may i have a table for four in london
L: yes please.	S: any preference on a type of cuisine
T: burchak.	U: italian ... wait no no ... british please
L: okay, a sako burchak.	S: which price range are looking for
T: good job.	U: em... in a cheap price range please

T: simulated tutor *L: system* *U: simulated customer* *S: system*

TABLE 6.6: User Simulation Examples for the BURCHAK and the bAbi Corpora (a) on the BURCHAK corpus (Yu et al., 2017b) (b) on the bAbi corpus (Weston et al., 2015)

6.5 Chapter Summary

In this Chapter, we aimed at designing and implementing a user simulation to simulate the tutor behaviours from the BURCHAK corpus collected in the previous chapter. We briefly looked through a list of existing User Simulation and their relevant approaches for natural language processing, where we found that, although there have been a large number of user simulation applying to develop/test the dialogue strategies by resembling human behaviours in conversations, they rarely take into account simulating natural, complex daily human-human conversations, referring to a wide range of dialogue phenomena or disfluency (Hough and Schlangen, 2017) e.g. overlapping, self-correction, self-repetition, and continuation, which have attracted our attentions in this project. Hence, our model here is built to deal with the simulation of dialogue phenomena as an essential component. The implemented model contributes to the prediction of user models on multiple levels, including full utterances, dialogue actions, and a sequence of words. The most essential feature of this framework is that it is able to mimic the dialogue phenomena that occur in the realistic conversation in the corpus (see Chapter 5).

In this chapter, we evaluated our user models on two different dialogue corpus, i.e. BURCHAK and Facebook bAbi corpora. The results obtained from the evaluation indicate that:

- The user simulation for both data sets has shown good performance on multiple levels, especially on the dialogue action level. This might because the user simulation on the dialogue act level contains much more abstract but less variate responses than the other levels.
- The user simulation for the bAbi corpus outperforms the one for the BURCHAK, due to the fact that the bAbi corpus is a synthetic dialogue data set in which dialogues are more predictable with less variations.

We have proven that the proposed user simulation is able to produce coherent user responses from a realistic dialogue corpus on different tasks/corpus, e.g. the task of learning visual attributes and the task of booking restaurants. The framework will be applied to build a simulated tutor for training and testing new learning-oriented dialogue agents by teaching visual colours and shapes through natural, human-like conversations with the agent in further research.

As mentioned before, the BURCHAK corpus also contains massive variation on both dialogue strategies and capabilities in the realistic human-human conversations. In the next chapter, we will present several experiments to explore their actual effect on the learning performance by comparing the dialogue agent under different conditions (i.e. combinations of different dialogue strategies). We will also focus on discussing and investigating both the *Tutor*'s and the *Learner*s preferences and capabilities while interactively learning novel objects or attributes. This investigation is likely to directly contribute to the training and implementation of an efficient NL learning system (see Chapter 8).

Chapter 7

Effect of Dialogue Strategy on Interactive Learning/Grounding Tasks

In the previous chapter, we have introduced a collection of human-human dialogues (BURCHAK) that contains a variety of dialogue strategies and capabilities adopted by either the tutor or the learner on an interactive learning task. We believe that these dialogue features, especially strategies (e.g. *initiative*, *uncertainty*, *context-dependency* and *knowledge-acquisition*), may not only lead to different behaviours of all interlocutors (including human tutors and a learner system), but also impact on the final learning performance of the learner (i.e. how well the learner can learn all visual attributes at the end of conversation). In this chapter, we mainly focus on investigating the possible effects of these dialogue strategies on the interactive learning task. We hypothesize that *an agent which takes initiative in dialogue, considers uncertainty, context-dependency, and knowledge-acquisition can achieve a better learning performance than other agents which do not consider these features.*

Given an interactive learning task in this project, a good learning agent may be defined as a smart system that is able to learn novel visual knowledge by effective dialogue strategies, i.e. actively asking for useful information, instead of passively waiting for feedback from tutors. More specifically, the agent should learn to control human involvement in the learning task, i.e. get more information but with less tutor help. For example, in the beginning of the learning task, since the agent does not have any knowledge about the visual scene, it must wait for or ask the tutor to provide attribute words, "what is the colour of this?". But with more examples trained through dialogue, the agent is more likely to describe the particular object correctly without additional assistant from the tutor than before, like "okay, i think this is a red square. next object please.". Here, the human involvement on such interactive

learning task is defined by the effort needed by the tutor to teach visual knowledge through dialogue, which is called dialogue/tutoring cost.

In this chapter, we defined the overall performance of an interactive learning agent as a trade-off between the recognition performance and the human involvement on the dialogue level. In the other words, a good learning agent should be able to achieve higher recognition score but with less dialogue cost.

In order to prove the aforementioned hypothesis, in this Chapter, we briefly describe all dialogue strategies we considered in the task in Section 7.1. It is followed with two main experiments that explore a better dialogue/learning strategy by comparing different strategy combinations on the tutor side (Section 7.2) and the learner (system) side (Section 7.3) respectively. We evaluate each combination via both the recognition performance, the dialogue cost, and their trade-offs.

7.1 Diverse Dialogue Strategies for Interactive Learning

In this section, we aim at exploring a solution that leads to an excellent trade-off between classification performance of an agent and learning/dialogue effort by tutors, i.e. a learning agent learns to identify novel object as accurately as possible, but with as little tutor involvement as possible. Here, we investigate a number of dialogue policies and capabilities that can contribute to different behaviours for interactively learning knowledge. These dialogue policies can be identified from two aspects: *tutor-based strategy* and *learner-based strategy*. The former strategies can be applied to discover how tutor's behaviours affect responsiveness and learning capacity of the learner. In contrast, investigations of the latter one is to mainly study effectiveness of several dialogue factors through an interactive learning process. We will discuss these strategies separately in the following sections.

7.1.1 Dialogue Capabilities

Given the statistical analysis of the BURCHAK corpus, a list of basic capabilities, such as *inform* and *WH-question*, *polar-question*, *demand*, *confirmation* as well as *correction*, contribute to natural conversations for the interactive learning task on both sides of the tutor and the learner. These capabilities support the participant to acquire/present necessary and correct information (visual-attribute words) through interaction with each other. They are listed with brief explanations as below:

- **Inform**: called by both the tutor and the learner, may describe visual attributes of a specific object, for instance "T: it is a (red) square".
- **WH-Question**: applied by both tutor and learner to directly gain information for a particular visual object or attribute by asking a *WH* question, e.g. "T: what can you see here?".
- **Polar-Question**: happens on both sides of the tutor and the learner, while the tutor wants to test the learner's knowledge or the learner has a lower confidence about its answers, e.g. "is this a red square?"
- **Demand**: applied only by the learner while the learner cannot receive enough information about a specific visual object/attribute, normally occurs in the form of the information-asking, e.g. "T: what colour/shape is this?".
- **Implicit/Explicit Confirmation**: (only by the tutor), processes acceptance or rejection on the learner's statements or answers, for example, "yes, it's a blue square / no, it isn't.".
- **Correction**: called by the tutor only, processes corrections by presenting correct information/labels, normally occurs with the capability of rejection, e.g. "T: no, it is blue.".

In addition, we define a "*listen*" capability, in which either tutors or learners will keep silence without interaction and release the floor to the others. The capability of "*listen*" allows: 1) the *learner* to directly update its memory while the *tutor* is making a statement about a specific object/attribute; or 2) the ***tutor*** to ignore any questions or statements from the learner, where the learner cannot update their representations immediately. On the other hand, the capability of ***Statement*** involves into two dimensions in dialogues:

1. ***Describing certainty***: allows the tutor/learner to make a statement about a particular object while they have high confidence on classifier predictions.
2. ***Expressing uncertainty***: allows the tutor/learner to may describe a particular object with uncertainty while having low confidence, e.g. "This is probably/maybe a red square."

These capabilities will be applied in all dialogue strategies (as discussed below) in the rest of this chapter.

7.1.2 Tutor-based Dialogue Strategies

This section defines the tutor's dialogue behaviours on learning tasks. Following the previous work (Skočaj et al., 2009), we generally identify tutors' behaviours into two groups, as below:

1. ***Tutor-Driven (TD)***: ***Tutor-Driven (TD)***: The tutor always gives available information about a particular object, i.e. supervised learning (always providing labels), by directly making statements (e.g. "this is a square" or "this is a red square"). This means that the whole learning process is an unidirectional interaction only handled by the tutor. In this case, the learner only needs to listen and update its learning models (the visual classifiers) upon what information the tutor presented.

2. ***Tutor-Corrected (TC)***: while the learner is describing or asking something about the object, the tutor only asks WH questions and corrects mistakes of the learner, and otherwise confirms correct statements (e.g. "T: what is this? L: this is a red square. T: yes/no, it is a green square" in Fig. 7.1). In contrast to the TD behaviour, the learner performs more actively to get involved with the learning process with its own predictions/knowledge. It will update its classifiers only when the tutor provides answers or confirms.

	TD	TC (-UC)	TC (+UC)
Good(Ideal) Tutor	**T:** this is red. **L:** okay. **T:** this is a square. **L:** okay. ---------- or ---------- **T:** this is a red square. **L:** okay.	**T:** what is this? **L:** this is a red square. **T:** no, this is a green square. **L:** okay.	**T:** what is this? **L:** is this a red square? **T:** no, this is a green square. **L:** okay.
Lazy(natural) Tutor	Without Knowledge-Demanding (-KD)		
	T: this is red. **L:** okay.	**T:** what is this? **L:** this is a red square. **T:** yes, it is a square. **L:** okay.	**T:** what is this? **L:** is this a red square? **T:** yes , this is a square. **L:** okay.
	With Knowledge-Demanding (+KD)		
	T: this is red. **L:** okay. what shape is it? **T:** this is a square. **L:** okay.	**T:** what is this? **L:** this is a red square. **T:** no, this is a circle. **L:** okay. Is the colour correct? **T:** yes. **L:** okay.	**T:** what is this? **L:** is this a green circle? **T:** no, this is a square. **L:** okay. Is the colour correct? **T:** no, this is red. **L:** okay.

FIGURE 7.1: Examples Dialogues in Different Tutor-based Behaviours

According to the previous work from Skočaj et al. (2009), both tutor behaviours are frequently adopted in a perceptual learning process, which may lead to different levels of learner involvement. They assumed that the tutor can always perform well through the entire learning process. However, this may be idealised for real-world problems, in which human tutors

may not always supply enough information about a certain visual object. In this thesis, we therefore also take the following situations into account:

- ***"Good-Tutor" (GT)***: the tutor always gives all labels for each visual object, always corrects all the mistakes of the learner, and always confirms correct statements by the learner.
- ***"Lazy-Tutor" (LT)***: this tutor only gives one of the correct labels at a time (e.g. "it's red" or "it's a square"), and only corrects one mistake at a time. It always confirms when asked to. This tutor is more similar to what we can expect from real human behaviour when teaching robots than the Good Tutor.

In real-world learning tasks, a learner might be required to consider several additional capabilities, which may support to respond to tutor behaviours in a natural way, especially within a *Lazy-Tutor* situation. Moreover, these capabilities are also likely to help improve the overall performance of the learner at the end of learning process, i.e. achieving a better trade-off between the performance of object/attribute recognition and the dialogue cost for the tutor. In this thesis, we carry out experiments with two core dialogue strategies with binary levels:

1. **Uncertainty (+UC/-UC)**: determines whether the learner takes into account, in its dialogue behaviour, its own subjective confidence about the attributes of the presented object. The confidence is the probability assigned by any of its attribute classifiers of the object being a positive instance of an attribute (e.g. 'red') - see below for how a confidence threshold is used here in section 7.1.4. In +UC, the agent will not ask a question if it is confident about the answer, and it will hedge the answer to a tutor question if it is not confident, e.g. "T: What is this? L: is this a red square?". In -UC, the agent always takes itself to know the attributes of the given object (as given by its currently trained classifiers), and behaves according to that assumption.
2. **Knowledge-Demanding (+KD/-KD)**: so-called knowledge-acquisition (explained in Chapter 5, which determines whether the learner can request further details/information about objects, which may be useful when interacting with a "Lazy" Tutor (described above). In condition +KD, the learner is able to request more information by asking extra questions (see Fig. 7.1 e.g. "what (colour/shape) is it? or "is the colour correct?". Otherwise, the learner with -KD will only update the classifiers based on the information provided.

7.1.3 Learner-based Dialogue Strategy

In contrast with the previous section, the goal of this section is to investigate learner's behaviours in learning process through dialogue with human tutors. In general, based on the tutor's behaviours, there are several different dialogue capabilities and policies from the learner that a continuous concept-learning system might have or adopt. These may lead to different outcomes for the accuracy of the learnt concepts/meanings, learning rates and cost to the tutor – as well as with trade-offs between these. Apart from the dialogue capability of **Uncertainty** ***(+UC/-UC)*** as described above, we identify two additional dialogue factors that also determines the learner's dialogue behaviours, **Initiative** ***(T/L)*** and **Context-Dependency** ***(+CD/-CD)*** as described in Chapter 5:

1. **Initiative** (**L**earner/**T**utor): determines who takes initiative in the dialogues. When the learner has initiative, it is the one that drives the conversation forward, by making a statement about the attributes of the object, asking questions to the tutor for information or confirmation (e.g. "What colour/shape is this?" or "Is this red?"), initiates topics etc. On the other hand, when the tutor takes initiative, the learner will wait for the tutor until he/she starts to ask questions to the learner (e.g. "What colour is this?" or "So this is a ....") or making a statement about the attributes of the object.

2. **Context-Dependency** (**+CD**/**-CD**): determines whether the learner can process (produce/parse) context-dependent expressions such as short answers and incrementally constructed turns, e.g. "T: What is this? L: a square", or "T: So this one is ...? L: red/a circle". This is a setting that can be turned off/on in the DS-TTR dialogue model.

Fig. 7.2 shows example interactions between the learner and the tutor in some of the experimental conditions. Noting how the system is able to deal with (parse and generate) utterance continuations as in $T + UC + CD$, short answers as in $L + UC + CD$, and polar answers as in $T + UC + CD$.

7.1.4 Confidence Threshold

Regarding to addressing the uncertainty with both tutor- and learner-driven strategies, a confidence threshold is introduced. To determine when and how the agent properly copes with its attribute-based predictions, we use confidence-score thresholds. It consists of two values, a base threshold (e.g. 0.5) and a positive threshold (e.g. 0.9).

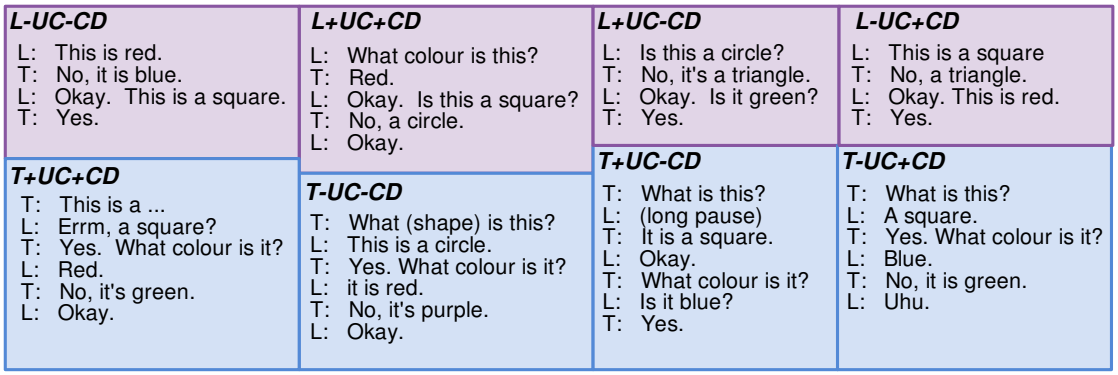


FIGURE 7.2: Examples Dialogues in Different Conditions

If confidence scores of all classifiers are under the base threshold (i.e. the learner has no attribute label that it is confident about), the agent will ask for object information directly from the tutor via *WH-questions* (e.g. "L: what is this?").

On the other hand, if one or more classifiers score above the base threshold, then the positive threshold is used to judge to what extent the agent trusts its prediction or not. If the confidence score of a classifier is between the positive and base thresholds, the learner is not very confident about its knowledge, and will check with the tutor, e.g. "L: is this red?". However, if the confidence score of a classifier is above the positive threshold, the learner is confident enough in its knowledge not to bother verifying it with the tutor. This will lead to less effort needed from the tutor as the learner becomes more confident about its knowledge. However, since a learner with high confidence will not ask for assistance from the tutor, a low positive threshold may reduce the chances that allow the tutor to correct the learner's mistakes.

We therefore tested different fixed values for the confidence threshold and this determined a fixed 0.5 base threshold and a 0.9 positive threshold were deemed to be the most appropriate values for an interactive learning process - i.e. these values preserved good classifier accuracy while not requiring much effort from the tutor.

7.2 Experiment 1: Effects of Tutor-based Dialogue Strategies on the Learning Performance

As described in the beginning of this chapter, we aims at exploring a suitable learning/-dialogue strategy that can help the agent achieve higher learning/recognition performance but also with less human involvement. The experiment presented in this section therefore

is designed to investigate how different combinations of these tutor-based strategies can affect the overall performance of the agent in an interactive learning process. Generally, We evaluate the performance of these dialogue strategy combinations from three aspects: (1) recognition performance, (2) human involvement on the dialogue level, and (3) a relative balance between (1) and (2). More details are presented as below:

7.2.1 Visual Object DataSet

Here, we build up a new visual dataset consisting of 600 images of simple handmade objects. The goal of this system was to learn simple visual attributes (e.g. colour and shape) from these simple objects (see Fig. 7.3). There are nine attributes considered in this dataset: 6 colours (e.g. black, blue, green, orange, purple and red) and 3 shapes (e.g. circle, square and triangle). All images are annotated with these 9 attribute labels by ourselves. As background noise might interfere with the ability of object segmentation and extraction, we build images containing only one object with a white background. The system can then automatically detect object boundaries and build the corresponding perceptual representations as described in Chapter 3, i.e. a combination of HSV colour space for colour attributes and bag-of-visual-words for shape.

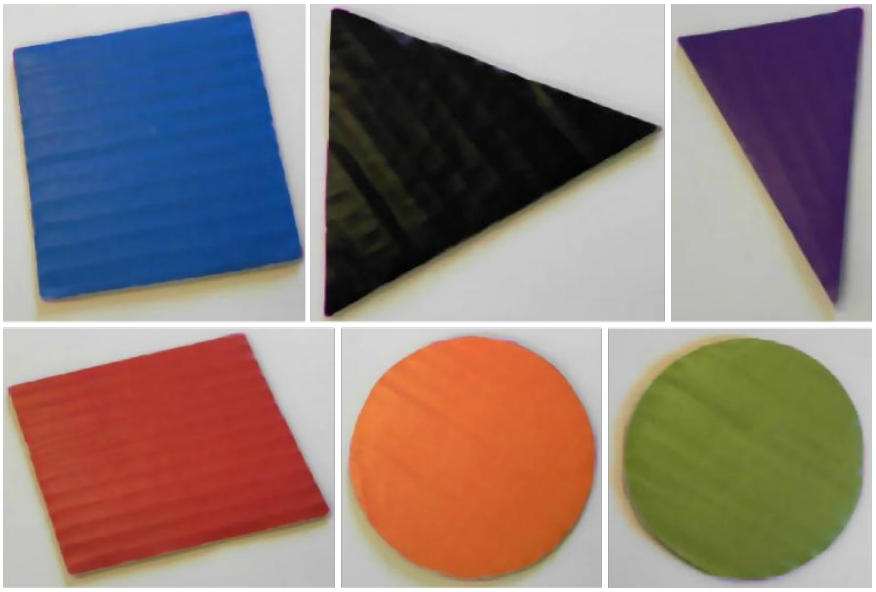


FIGURE 7.3: Examples of simple handmade objects

To ensure that each attribute-based classifier can be learned with enough positive examples, we kept a relative balance between different attributes in the same categories, i.e. 100 positive instances of each colour property, 147 instances of "circle", 232 instances of "square", and 221 instances of "triangle".

7.2.2 Task & Procedure

In this section, the main goal of this section is to investigate effects of different dialogue strategies from tutor-driven aspects on the learning performance by comparing them on an interactive learning task. Through this experiment, a set of rule-based learning agents with different combinations of tutor-based strategies are required to learn novel visual concepts/attributes (colour and shapes) of different objects through interaction with a simulated tutor (see Chapter 6). During the learning period, the agent will randomly go through 500 visual instances from the visual dataset described above, each instance for a single dialogue, the rest of this dataset (around 100 images) are applied to measure the learning performance (recognition score) of classifiers. In this experiment, we performed a 20-fold cross validation.

More specifically, we compare different behaviours and capabilities with two baseline policies without corrections (NC), in which the learner cannot process corrections but only confirmations from the tutor. This means that the learner can only update its classifiers when its own predictions are correct. On the other hand, we applied the constant confidence thresholds (with a base threshold (0.5) and a positive threshold (0.9)) to describing uncertainty described above. It means that if the confident score of the classifier is higher than the positive threshold, there is no conversation, i.e. the agent will train its classifiers without asking feedback from the tutor at all.

Besides, we also define a **Learning Step** as comprised of 25 such dialogues[1]. At the end of each learning step, the system is tested using the test set. The values used for measuring the overall performance of each combination of dialogue strategies at each learning step.

7.2.3 Evaluation Metrics

Following the PARADISE evaluation framework by Walker et al. (1997) for task-oriented dialogue systems, we consider a good performance of the learning agent with two key metrics: 1) *Recognition score* that measures how well the agent can recognise visual attributes after dialogues, and 2) *Dialogue cost* (human involvement) that measure how much human tutors need to involve into the learning task on the dialogue level. The best agent should be able to achieve and remain a better trade-off between the learning performance and the dialogue cost, i.e. higher learning performance but with less dialogue cost.

Recognition score is a metric that measures the overall accuracy of the learnt word meanings / classifiers, which "rewards successful classifications (true positives and true negatives) and penalizes incorrect predictions (false positives and false negatives)" Skočaj

[1]We also attempt to define the learning step with less dialogues, e.g. 10, 15, and 20 dialogues, with less training dialogues, the agent cannot show significant differences between every two learning steps.

et al. (2009). As the proposed system considers both correctness of predicted labels and prediction confidence on learning tasks, the measure will also take the true labels with lower confidence into account, as shown in Table 7.1: "LowYes" means that the system made positive predictions but with lower confidence. In this case, the system can generate a polar question for requesting tutor feedback. "LowNo" is similar to "LowYes", but only works on negative predictions.

		Predicted Labels			
		Yes	LowYes	LowNo	No
Actual Label	Yes	1	0.5	-0.5	-1
	No	-1	-0.5	0.5	1

TABLE 7.1: Recognition Score Table

Dialogue/Tutoring Cost Here, we introduce a metric called *dialogue cost* that measures how much a human tutor involved into the learning task on the dialogue level. In the other words, it reflects the effort needed by a human tutor in interacting with the system. Skočaj et al. (2009) point out that a comprehensive teachable system should learn as autonomously as possible, rather than involving the human tutor too frequently. There are several possible costs that the tutor might incur: C_{inf} refers to the cost of the tutor providing information on a single attribute concept, e.g. "this is red" or "this is a square" (assigned to 1); C_{ack} (around 0.25) is the cost for a simple confirmation (like "yes", "right") or rejection (such as "no"); $C_{crt}$is the cost of correction for a single concept, e.g. "no, it is blue" or "no, it is a circle" (1). We associate a higher cost with correction of statements than that of polar questions. This is to penalise the learning agent when it confidently makes a false statement – thereby incorporating an aspect of trust in the metric (humans will not trust systems which confidently make false statements). Finally, we are also concerned with a tutoring cost for each dialogue turn C_{turn} (about 0.15).

We define the overall tutoring cost at particular learning steps as:

$$C_{dlg} = \sum_{i=1}^{n} C_{inf} + \sum_{j=1}^{n} C_{ack} + \sum_{k=1}^{n} C_{crt} + \sum_{p=1}^{n} C_{turn} \tag{7.1}$$

Overall Learning Performance In this experiment, we mainly consider the overall performance of a learning agent as a trade-off between the learning performance (recognition score S_{recog}) and the human involvement (dialogue cost C_{dlg}), as formulated below:

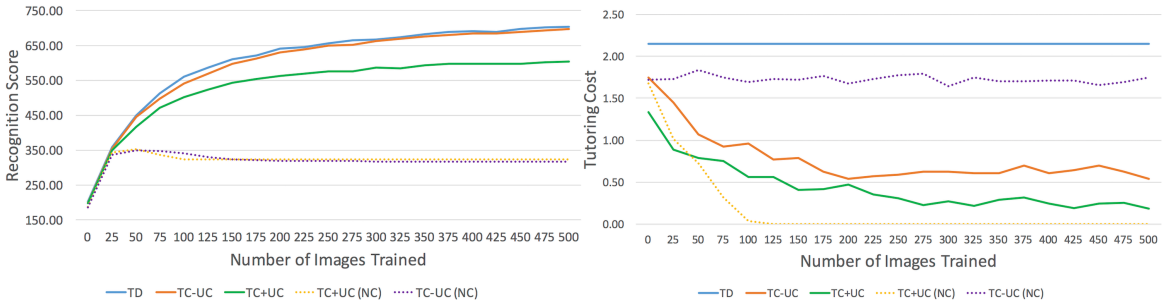
$$Overallperformance = \frac{\Delta S_{recog}}{C_{dlg}},$$

where ΔS_{recog} represent the difference of recognition score between every two most closed learning steps; C_{dlg} represents a dialogue cost for each individual episode (dialogue).

In the experiment, we compare the learner's overall performance across the different dialogue strategy conditions. We seek dialogue strategies that maximise this overall performance.

7.2.4 Results & Discussion

Figures 7.4 and 7.5 plot the learning performance (the recognition scores and dialogue cost) of different tutor-based strategies in the good and lazy tutoring situations respectively. In these figures, x-axis represents the number of visual instances that have been learned in the learning process, y-axis in Figs. 7.4a and 7.5a represent the recognition scores of classifiers at each learning step, but y-axis in Figs. 7.4b and 7.5b represent the average tutoring/dialogue cost across 25 dialogue in each single learning step[2].



(A) Recognition Score (B) Tutoring Cost

FIGURE 7.4: Evolution of Learning Performance in the *Good Tutor* Condition (TD = tutor-driven, TC = tutor-corrected, +/-UC = with/without the learner ability of processing uncertainty, NC = no correction process ability)

Condition	Recognition Score	Tutoring Cost
TD	702.975	2.500
TC-UC	699.800	0.779
TC+UC	590.825	0.407

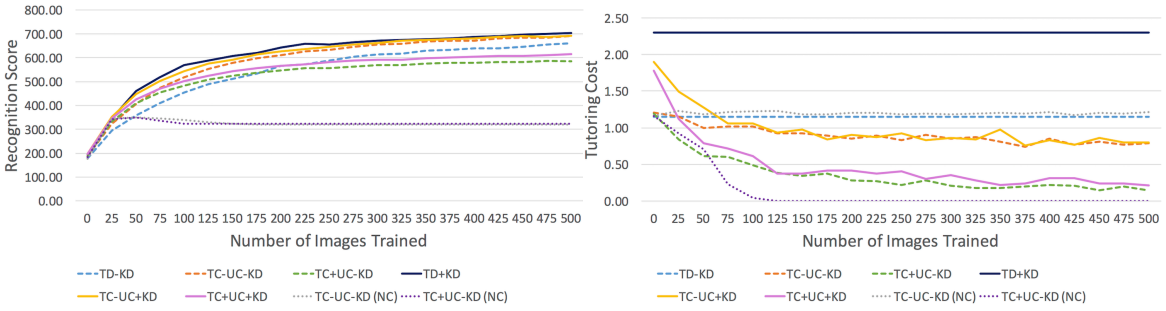
TABLE 7.2: Table of average Recognition Score and Cost under Different Conditions for a "Good" Tutor

Here, we firstly investigate the improvement of learning performance over time for different learner policies and capabilities with an ideal tutoring situation (*Good* Tutor). We compared both tutor type (TD and TC) with corresponding learner strategies and capabilities (+/-UC and NC) in terms of Recognition Score and Tutoring Cost. (Note that in the Good Tutor case, +/-KD has no effect). A between-subjects Analysis of Variance (ANOVA) shows that both Tutor Type ($p < 0.01; F = 1981.47$) and Uncertainty ($p < 0.01; F = 82.846$) have significant effects on the dialogue/tutoring cost through the interactive learning task, but

[2]Here we plot the average dialogue cost for each learning step (across 25 dialogues, the value will drop down when the classifiers are trained with more and more visual instances, because there will be fewer or no interactions between the tutor and the agent once it can correctly recognise the visual attributes)

only Uncertainty ($p < 0.01; F = 122.329$) significantly affect the recognition performance of the agent. We present the mean recognition score and cost under different conditions in Table 7.2.

Fig. 7.4a shows that the Tutor-Driven (TD, blue line)and Tutor-Corrected without Uncertainty (TC-UC, red line) conditions achieve the highest Recognition scores, but ones without the Learner ability of processing corrections (NC) perform badly, as expected. In terms of Tutoring Cost though, we see that TD has a high cost while TC-UC has quite low cost. Interestingly, TC+UC (Tutor-corrected, with Uncertainty, green line), has a lower cost than both of these conditions, while still achieving a high Recognition score. This is because the Learner which is aware of its uncertainty about classifier outputs requires fewer corrections from the Tutor, while the classifiers still become more accurate over time.



(A) Recognition Score (B) Tutoring Cost

FIGURE 7.5: Evolution of Learning Performance in the *Lazy Tutor* Condition (TD = tutor-driven, TC = tutor-corrected, +/-UC = with/without the learner ability of processing uncertainty, +/-KD = with/without Knowledge-demanding ability, NC = no correction process ability)

Similar to Fig. 7.4, Figs. 7.5a, b show the Recognition Score and Tutoring Cost respectively for the same learner strategies, but with a more natural tutoring situation (*Lazy* Tutor), and where the learner can be Knowledge-Demanding (+/-KD). Similar to the *Good* Tutor experiment above, the ANOVA test shows significant main effects of Tutor Type ($p < 0.01; F = 148.205; MeanSquare = 12.444$), Uncertainty ($p < 0.01; F = 547.273$), as well as Knowledge-demand ($p < 0.01; F = 652.388$) on the tutoring cost of the agent. Besides, both Uncertainty ($p < 0.01; F = 156.507$) and Knowledge-demand ($p < 0.01; F = 8.765$) have significant impacts on the learning performance. We also present the mean recognition score and cost under different conditions for the "'Lazy" tutor in Table 7.3.

Given a *Lazy* tutor, both the TD and TC-UC policies, without Knowledge-demand (-KD), show slightly worse recognition performance than they did under the Good Tutor policy, because the learner does not gain as much knowledge from the tutor in each single dialogue turn. Whilst both policies cost much less than before for the same reason, they show better performance in the tutoring cost (as compared between Figures 7.4b and 7.5b). By contrast,

Condition	Recognition Score	Tutoring Cost
TD-KD	660.365	1.150
TD+KD	703.031	2.300
TC-UC-KD	695.475	0.885
TC-UC+KD	686.675	0.987
TC+UC-KD	583.325	0.361
TC+UC+KD	595.975	0.422

TABLE 7.3: Table of average Recognition Score and Cost under different conditions for a "Lazy" tutor

as a situation with two incorrect predictions rarely occurs with the TC+UC-KD policy (for only about 20 out of 500 images), the *Lazy-Tutor* policy will not affect Recognition Score or Tutoring Cost very much for the TC+UC policy (see Fig. 7.5a, b). Moreover, the results in Figure 7.5 also show that a Knowledge-Demanding (+KD) learner policy may always improve recognition performance (Fig. 7.5a).

For the aforementioned reasons, we consider the overall learning performance as a trade-off between the recognition score and dialogue cost. Hence, in the Good-Tutor condition, the TC-UC policy (orange line) shows better overall performance than TD (blue line) because of its comparable recognition score but lower tutoring cost. In addition, though the Uncertain Learner (TC+UC) policy performs slightly worse on recognition score (this might be due to insufficient error detection and recovery), it also reduces the tutoring cost through time in both good or lazy tutoring situations.

Since our ultimate goal here is to create a full dialogue system that can learn accurate concepts (word meanings) with little effort from human tutors, these results would lead us to choose a dialogue system that can can handle corrections – i.e. some variant of the Tutor Corrected system. The results show that, depending on the relative weight between Recognition Score and Tutor Cost, an optimal Learner Dialogue Policy could, for example, use TC-UC(NC) for the first 50 or 60 images, and then switch to TC+UC. We investigate such dynamic policies and their optimisation in a later study using Reinforcement Learning methods (see Chapter 8)

7.3 Experiment 2: Effects of Learner-driven Dialogue Strategies on the Learning Performance

In this experiment, we aims at investigating the effects of different combinations of learner-based dialogue strategies (as described above) on the overall learning performance of an

interactive agent. We takes into account two essential metrics to evaluate the overall performance, including learning accuracy and dialogue-level human involvement (dialogue cost). We expect to find a good dialogue strategy that can achieve a good trade-off between these two metrics, higher accuracy but with less dialogue cost.

7.3.1 Visual Data

Here, we keep making use of the visual dialogue data set described above that contains 600 images of simple handmade objects with 9 visual attributes 6 colours and 3 shapes. All images contains a unique visual object with the white background.

7.3.2 Task & Procedure

Similar to the previous experiment, here we aims at investigating the effects of learner-based dialogue strategies on the overall learning performance by comparing different combinations of learner-based strategies that learn unseen visual attributes from a simulated tutor through dialogue. We set a $2 \times 2 \times 2$ factorial experiment, with three factors (e.g. *initiative*, *uncertainty* and *context-dependency*) with two levels. Together, there factors will determine the learner's behaviour. Contrast with experiment with tutor-based strategies, the simulated tutor in this experiment keep constant teaching/dialogue behaviours across all conditions. Its policy is that of an always *truthful*, *helpful* and *omniscient* one: it (1) has complete access to the labels of each object; and (2) always acts as the context of the dialogue dictates: answers any question asked, confirms or rejects when the learner describes an object; and (3) always corrects the learner when it describes an object erroneously. In this learner-based experiment, each interaction episode about an object ends either when both the shape and the colour of the object are discussed and agreed upon, or when the learner requests to be presented with the next image (this happens only in the Learner initiative conditions).

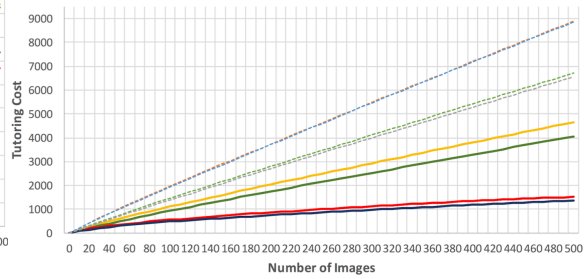
Note that, instead of 25 episodes, we here define the learning step as comprised of 10 such episodes.

7.3.3 Evaluation Metrics

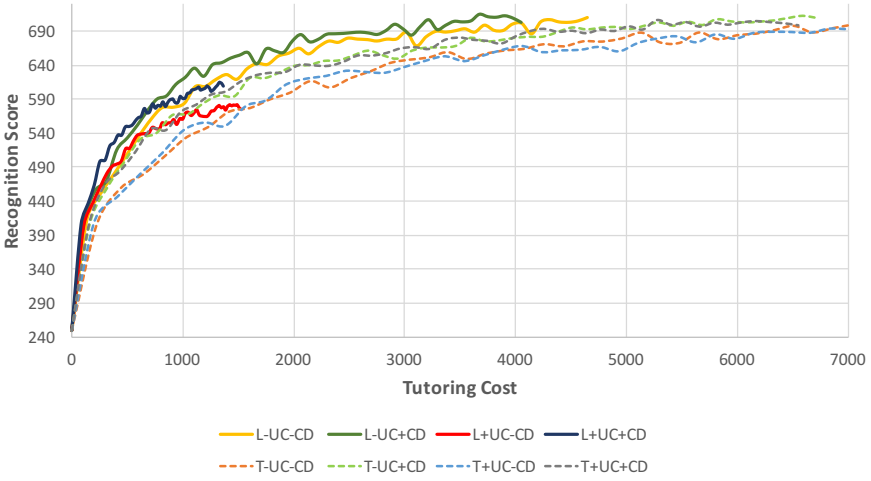
In this experiment, we loosely follow the evaluation metrics described in the previous experiment, i.e. measure the overall performance of an interactive system from both learning performance and dialogue cost.

However, in terms of the dialogue/tutoring cost, we keep most cost scores same as those in the previous experiment, see Table 7.4. However, instead of considering the cost for each

(A) Recognition Score



(B) Tutoring Cost



(C) Overall Performance

FIGURE 7.6: Evolution of Learning Performance

dialogue turn, here we takes into account parsing (C_{parse}) as well as production ($C_{production}$) costs for tutor: each single word costs 0.5 when parsed by the tutor, and 1 if generated (production costs twice as much as parsing). These exact values are based on intuition but are, of course, kept constant across the experimental conditions and therefore do not confound the results we report below.

C_{inf}	C_{ack}	C_{crt}	$C_{parsing}$	$C_{production}$
1	0.25	1	0.5	1

TABLE 7.4: Tutoring Cost Table

In this experiment, we still look for the most appropriate dialogue strategy combination that can achieve good trade-off between accuracy and dialogue cost.

7.3.4 Results & Discussion

7.3.4.1 Results

Figures. 7.6a and 7.6b plot the progression of Recognition Score and (cumulative) Tutoring Cost for each of the 8 conditions in this experiment, as the system interacts over time with the tutor about each of the 500 training instances. As noted in passing, the vertical axes in these graphs are based on averages across the 20 folds - recall that for Accuracy the system was tested, in each fold, at every learning step, i.e. after every 10 training instances.

Fig. 7.6c, on the other hand, plots Recognition Score against Tutoring Cost directly. Note that it is to be expected that the curves should not terminate in the same place on the x-axis since the different conditions incur different total costs for the tutor across the 500 training instances. This curve corresponds to the trade-off between Recognition Score and the Tutoring Cost during an interactively learning period. It constitutes our main evaluation measure of the systems overall performance in each condition, and it is this measure for which we report statistical significance results:

The ANOVA results show significant main effects of Initiative ($p < 0.01; F = 469.2$), Uncertainty ($p < 0.01; F = 179.8$) and Context-Dependency ($p < 0.01; F = 20.12$) on the system's overall performance, trade-off between the accuracy and the dialogue cost. There is also a significant Initiative×Uncertainty interaction ($p < 0.01; F = 181.72$). The average accuracies, costs and ratios under different conditions are shown in the Table 9.3

Condition	**Accuracy**	**Tutoring Cost**	**Ratio**
T-UC-CD	0.8758	8861.40	0.00009882
T-UC+CD	0.8650	6653.00	0.00012996
T+UC-CD	0.8992	8857.00	0.00010113
T+UC+CD	0.9075	6552.25	0.00013819
L-UC-CD	0.8883	4583.000	0.00019383
L-UC+CD	0.8792	4024.000	0.00021848
L+UC-CD	0.7650	1368.725	0.00055882
L+UC+CD	0.7775	1081.250	0.00057133

TABLE 7.5: Table of average Accuracy, Cost and Ratio under different Conditions

7.3.4.2 Discussion

Tutoring Cost As can be seen on Fig.7.6b, the cumulative cost for the tutor progresses more slowly when the learner has initiative (L) and takes its confidence into account in its behaviour (+UC) - the grey, blue, and red curves. This is so because *a form of active learning* is taking place: the learner only asks a question about an attribute if it isn't

confident enough already about that attribute. This also explains the slight decrease in the gradients of the curves as the agent is exposed to more and more training instances: its subjective confidence about its own predictions increases over time, and thus there is progressively less need for tutoring.

Recognition Score On the other hand, the L+UC curves (grey and blue) on Fig. 7.6a show the slowest increase in accuracy and flatten out at about 600 and 580 respectively. This is because the agent's confidence score in the beginning is unreliable as the agent has only seen a few training instances: in many cases it doesn't query the tutor or have any interaction whatsoever with it and so there are informative examples that it doesn't get exposed to.

Comparing the gradients of the curves on Fig. 7.6c shows that the overall performance of the agent on the trade-off measure is significantly better than others in the L+UC conditions (recall the significant Initiative $\times$ Uncertainty interaction). It achieves good recognition (over 550) but with much less cost units (only 1500 units) than the other conditions at the end of learning task.

Finally, the significant main effect of **Context-Dependency** on the overall performance is explained by the fact in the +CD conditions, the agent is able to process context-dependent and incrementally constructed turns, leading to less repetition, shorter dialogues, and therefore better overall performance.

To summaries, according to these discussions presented above, an agent that takes the initiative, considers both uncertainty and context-dependency is more desirable to achieve a good trade-off between learning performance and dialogue cost (human involvement) than other strategies. However, since taking into account the uncertainty strategy might lead to the situation that the agent learns fewer correct examples than the other. In the further work, we will try to learn an strategy with an adaptive confidence threshold, which is likely to address such problem and improve the recognition performance, see Chapter 8.

7.4 Chapter Summary

In this chapter, we investigate the effects of different dialogue strategies on an interactive learning task, by comparing the overall performance (trade-off between learning accuracy and cost) of the system under different conditions (i.e. combination of different dialogue strategies and capabilities). We expect a system that can learn to correctly identify as many visual concepts (word meanings) as possible, but with less human involvement on the dialogue level (dialogue effort/cost). Results presented in this chapter would lead us to

choose a combination of dialogue strategies that can be corrected by humans – i.e. some variant of the Tutor/Learner-driven dialogue strategies (e.g. Kennington et al. (2015), Roy (2002)):

- In terms of the *tutor-based behaviours*, the results show that, the fully supervised cases (TD) have a high cost for the Tutor, and equivalent final recognition performance can be reached with less effort when using a Tutor-Corrected (TC) dialogue policy where the Learner can process corrections in dialogue. Final Recognition performance is slightly less good with learners which take their own uncertainty into account (TC+UC), but they require much less effort from Tutors, resulting in better overall performance.
- In terms of *dialogue factors* for the learner-based policy, the results show that, to maximise the learner's performance, the agent needs to take initiative in the dialogues, take into account its confidence about its predictions, and be able to process natural, human-like dialogue.

Through these experiments, a learning agent which takes initiative and takes into account uncertainty shows significant impacts on the learning process and the corresponding overall performance. In the next Chapter, following above exploration results, we will build the first learning/grounding agent that is trained using Reinforcement Learning (see Chapter 8) by learning visual colours and shapes through natural conversations with a simulated tutor. The tutor is trained based on the realistic human-human dialogue data from the BURCHAK corpus (Chapter 5). The optimised learning agent, not only learns to interact with the simulated tutor as naturally, coherently as possible, but also finds a better trade-off between learning performance and the tutoring cost. For evaluating the trained agent, we will compare it with the other hand-crafted rule-based dialogue systems on the interactive learning task.

Chapter 8

An Optimised Learning Strategy on the Dialogue-Act Level for Interactive Grounding Tasks

The previous chapter has indicated that for achieving a better overall performance (i.e. higher learning accuracy but with less effort from the tutor), the learner/agent has to take the initiative, and take into account the prediction uncertainty in conversations with the tutor through the learning period. A form of *active learning* is taking place in this case: the learner will actively acquire useful information from human partners by asking *WH* or *polar* questions, only when he cannot be highly confident on his answers. Compared to a learning system with a constant confidence threshold described in the previous chapter, we hypothesize that *the learner/agent with a threshold that is able to change dynamically over time can achieve a better overall performance (trade-off between accuracy and cost) in the interactive learning task than the constant one.* This is because an increasing number of training examples is likely to make the confidence score itself more reliable through the learning process.

In this chapter, to test this hypothesis, we introduce an optimised teachable agent for incrementally learning novel, visual attribute words through natural interaction from the tutor. In the section 8.1, this optimised agent, as the first prototype system, is trained using Reinforcement Learning (RL) and the Markov Decision Process (MDP) model against a "well-behaved" tutor, which is built based on the cleaned-up version of the BURCHAK corpus (see Chapter 5). The agent is trained to cope with two sub tasks: 1) learning to determine whether and when to ask for help from the tutor according to its prediction confidence; and 2) learning to process natural, human-like conversations with the tutor, for instance, massive variations and potential grammatical mistakes (the latter frequently occurs from

non-native speakers). To evaluate the overall performance of the optimised strategy, we designed an experiment (see Section 8.2) to compare the RL-based policy with several partially hand-crafted conversational policies, in which the learned policy shows a significantly better trade-off between accuracy and cost than the others, as well as successfully process natural, human-like conversations with users.

Finally, section 8.4 concludes the experiment results and summaries of our optimised dialogue agent for interactive learning goals.

8.1 Action-level Dialogue Agent for Learning

In this section, we design and implement the first interactively teachable system that learns to optimise its learning and conversation strategies to learn novel visual knowledge (for instance, colour and shape) from human tutors with a better trade-off between learning accuracy and the cost to the tutor in the interactive learning process. For understanding and learning the groundings between symbols in Natural Language and aspects in the physical surroundings, we deploy the **Interactive Multi-modal Framework** (as proposed in Chapter 3)(see Fig. 8.1), consisting of two core modules: a vision module and a dialogue module. The former module produces visual attribute predictions, using two base feature categories, i.e. the HSV colour space for colour attributes, and a 'bag of visual words' (i.e. PHOW descriptor) for the object shapes/class. It deploys a set of binary classifiers using Logistic Regression SVM classifiers with Stochastic Gradient Descent (SGD-SVM) (Zhang, 2004) to incrementally learn attribute predictions.

On the other hand, the latter one (Dialogue module) deploys a standard dialogue system, composed of Dialogue Management (DM), Natural Language Understanding (NLU) and Generation (NLG) components. The NLG implements a template-based utterance selector that chooses a suitable learner utterance for a specific dialogue act according to the statistical distribution of learner utterance templates from the BURCHAK corpus. The NLU employs a simple pattern-matching algorithm, called SimpleSLU [1]. The DM, as one of the most important components, relies on an optimised policy that is learned using RL (see the following section). The policy is trained to handle natural interactions with humans and to produce coherent dialogues and optimise the trade-off between accuracy of visual classifiers and the cost of the dialogue to the tutor. These components in the framework interact with each other via Dialogue Act representations (see more details in following sections).

[1] available at https://github.com/yy147/SimpleSLU

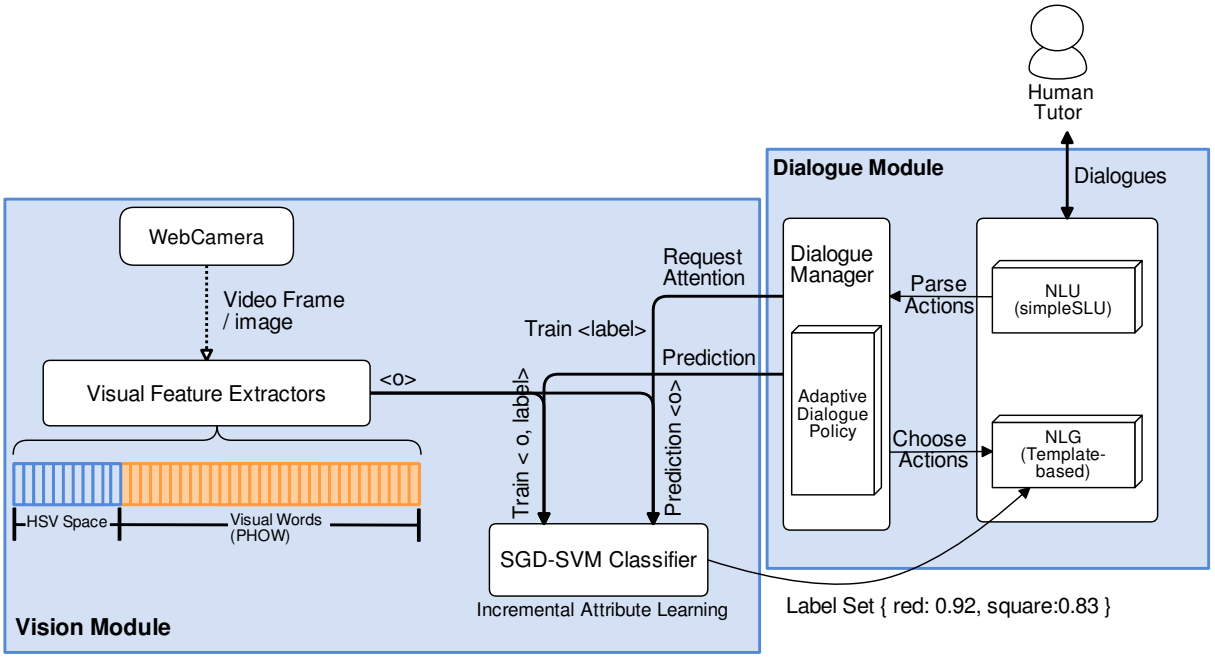


FIGURE 8.1: Multi-modal System Architecture integrated with Standard Dialogue System

8.1.1 Dialogue Act Tagging for Language Understanding: SimpleSLU

In this section, we present a simple Dialogue Act Tagging model, called Simple Spoken Language Understanding (SimpleSLU), which maps utterances to Dialogue Acts (DAt) (Stolcke et al., 2000): abstract meaning representations which specify what action the utterance is performing, as well as parameters of that action. This model performs a turn-level parse of the users utterance following a set of pre-defined rules, without considering the previous dialogue context. It searches for a sequence of intent-based patterns: slot-types and slot-values, and then produces a single dialogue act representation by packaging these key patterns (see more details in Chapter 3). Given the SimpleSLU tagging model, the visual classifiers in this section directly ground visual attribute words, such as 'red', 'circle', etc., that appear as parameters of the Dialogue Acts used in the agent, such as "inform(colour=red)", "polar(shape=square)".

As investigated in human-human conversations (see Chapter 5), a single dialogue turn may consist of splitting utterances, each of which presents different specific Dialogue Act, for instance in the interactive learning task, "Learner: I saw a square. Tutor: yes, it is. Now let's talk about its colour? what is the colour?", where it is composed of three actions: the tutor *acknowledges* learner's statement, and *move the topic* onto the colour attribute, and then *ask* for the colour. In such a learning process, every single action may affect on the understanding quality and even the task success. Therefore, SimpleSLU is also designed to cope with such multi-action issue (see Fig. 8.2).

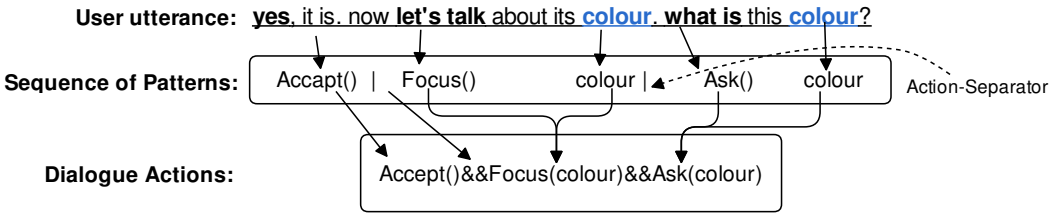


FIGURE 8.2: Multi-action Dialogue Processing with SimpleSLU model (where an action-separator (|) will be automatically detected upon special symbols, like dots and question-marks, excluding commas)

Similar to the single action detection described above, the model captures a sequence of key patterns following the word sequences of an utterance, and then converts them into dialogue acts, ontologies and entities (concepts). The model packages each Dialogue Act representation by searching for the nearest ontologies or entities which are still available (the searching path is always from the root (where the sequence starts) to the end). All actions are automatically separated by either an action-separator, which is detected upon special symbols (like dots and question-marks), or whether there are more than one action detected in a row. All actions in a turn are liked together with a special symbol &&. These actions are applied as DM input to predict the next appropriate system response, as shown below.

8.1.2 Dialogue Management: Optimised Learning Strategy with Multi-objective MDP

In this section, given the visual-attribute learning task, we implement a DM that processes a natural, coherent conversation with human tutors. We learned a learning/dialogue policy on the task of interactively learning novel visual attributes (e.g. colour and shape) for the DM. As discussed previously, given such an interactive learning/grounding task, a smart agent must learn novel visual objects/attributes as accurately as possible through natural interactions with real humans, but meanwhile it should attempt to minimise the human involvement as much as possible in this life-long learning process. This learning task can be formulated into two sub-tasks – *when* and *how* to learn (see Fig. 8.3) – which are trained using Reinforcement Learning with a multi-objective Markov Decision Process (MDP), consisting of two interdependent MDPs:

where the initial state of the Dialogue MDP is determined by the classifier prediction scores as well as chosen action from the Optimised Threshold MDP.

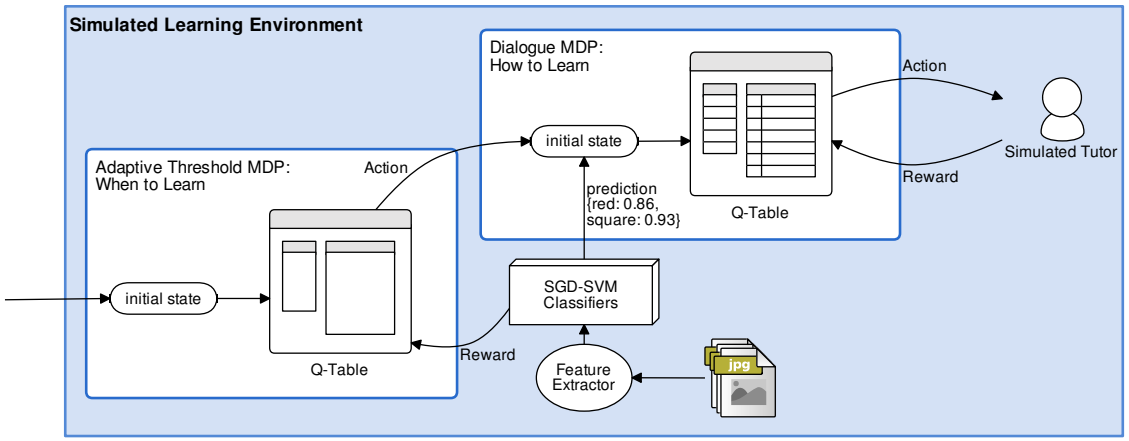


FIGURE 8.3: Optimised Learning Agent in Simulated Learning Environment

8.1.2.1 When to Learn: Optimised Confidence Threshold

In the first MDP, we expect the policy to learn when to acquire useful information from human tutors, where a form of *active learning* is taking place: the learner only asks a question about an attribute if it isn't confident enough already about that attribute. Following the previous work described in Chapter 7, we keep using the positive confidence threshold – a threshold which which determines when the agent believes its own predictions. This threshold plays an essential role in achieving the trade-off between the learning performance and the tutoring cost, since the agent's behaviour, e.g. whether to seek feedback from the tutor, is dependent on this threshold.

Here, we learn an adaptive strategy that aims at maximising the overall learning performance simultaneously, by properly adjusting the positive confidence threshold in the range of 0.65 to 0.95. We train the optimization as follows, in detail:

State Space The adaptive-threshold MDP initialises a 3-dimensional state space defined by $Num_{Instance}$, $Threshold_{cur}$, and $deltaAcc$, where $Num_{Instance}$ represents how many visual objects/images have been seen (the number of instances will be clustered into 50 bins, each bin contains 10 visual instances); $Threshold_{cur}$ represents the positive threshold the agent is currently applying; and $deltaAcc$ represents the level of changes of the classifiers recognition accuracy compared to the previous learning step (after seeing each 10 instances) – whether the classifier accuracy increases, decreases or keep constant comparing to the previous bin. The $deltaAcc$ is configured into three levels, as below:

$$deltaAcc = \begin{cases} 1, & \text{if } \Delta Acc > 0 \\ 0, & \text{else if } \Delta Acc = 0 \\ -1, & \text{otherwise} \end{cases} \tag{8.1}$$

, where the ΔAcc is measured on the separated collection of test instances.

Action Selection. Based on the previous performance of the classifier ΔAcc, the policy will update the current state space by either increasing or decreasing the current confidence threshold by 0.05, or keeping it constant.

Reward signal. The reward function for this sub-task is given by a local reward signal R_{local}, which is also directly proportional to the agents delta accuracy ΔAcc over the previous *learning step* (10 training instances, see above).

Each single training episode will be terminated once the agent goes through 500 instances.

8.1.2.2 How to Learn/Interact: Natural Interaction Dialogue Control

The second MDP aims at training an optimised dialogue policy that 1) learns how to effectively acquire important information from humans, and 2) learns to handle natural, human-like conversations from human tutors. At the end of a training process, the trained policy is expected to determine how to interact with human partners based on its visual prediction results: if the agent has a low confidence about its predictions (all confidence scores in the same attribute category are lower than 0.5), it may ask *WH* questions to directly acquire correct attribute words from humans, e.g. "what is this object?", otherwise the agent is able to make a guess about its answer by asking a yes-no question, for instance, "is this a red triangle?". In addition, the agent is also required to produce coherent conversations with a human partner – understand the particular dialogue utterances from humans and properly produce the next response.

Here, in order to achieves these goals, the Reinforcement Learning process and the corresponding MDP has been configured, as follows:

State Space. The dialogue agent initialises a 4-dimensional state space defined by (C_{state}, S_{state}, $preDAts$, $preContext$), where C_{state} and S_{state} are the status of visual predictions for the colour and shape attributes respectively (where the status is determined by the prediction score ($conf.$) and the adaptive confidence threshold ($posThd.$) described above (see Eq.8.2)), the $preDAts$ represents the previous dialogue actions from the tutor response, and the $preContext$ represents which attribute categories (e.g. colour, shape or both) were talked about in the context history.

$$State = \begin{cases} 2, & \text{if } conf. \geq posThd \\ 1, & \text{else if } 0.5 < conf. < posThd. \\ 0, & \text{otherwise} \end{cases} \tag{8.2}$$

where C_{state} or S_{state} will be updated to 2 also when the related knowledge has been provided/proved by the tutor.

Action Selection. The actions were chosen based on the statistics of the dialogue action frequency obtained from the BURCHAK corpus, including *question-asking(for WH questions or polar questions)*, *inform*, *acknowledgement*, as well as *listening*. These actions can be applied to either specific single attribute or both. The action of *inform* can be separated into two sub-actions – *polar question* while the agent is uncertain of its visual prediction results ($0.5 < conf. < posThd.$), otherwise *doNotKnow*.

Reward signal. The reward function for the learning tasks is given by a global function R_{global}, as below:

$$R_{global} = 10 - C_{ost} \tag{8.3}$$

where C_{ost} represents the cumulative cost by the tutor (see more details about the *Tutoring/Dialogue Cost* in Section 8.2.3) in a single dialogue.

Bias Reward Function: for learning a coherent conversation with the tutor, here we introduce a bias reward – Surprisial Probability ($P_{surprise}$) – which measures "the information surprise that is contained in data in an observer-dependent way related to all changes in expectation" (Baldi, 2002). Given the realistic dialogue on a learning task (see the BURCHAK corpus in Chapter 5), the probability of surprise for the next dialogue action from the learner (DAt_L) is calculated for a given observer (i.e. the last action executed by the tutor (DAt_T)) (see Eq. 8.4). The user simulation (as presented in Chapter 6) creates a surprisial dictionary that contains distributional probabilities across all possible dialogue action by the learner given a particular tutor action, where the higher surprisial value the action has, the more frequent the action was taken following the conditional action in the real data.

$$P_{surprise} = P(DAt_L \mid DAt_T) \tag{8.4}$$

Regarding the employment of this bias reward in the RL model, it is not directly applied in the global function, instead the surprisial probability determines whether the current

conversation in the training process will continue or not. If the surprisial probability of the chosen action is greater than 0, the model will continue the dialogue with the user simulation, otherwise the dialogue will be terminated immediately, not update the classifier on given labels, and return a large penalty of 1000. Meanwhile, the model will restart a new conversation on the same visual instance on the conversation that was broken down before. This setup of bias reward can help learn coherent conversation with the tutor from the realistic data, instead of applying a set of hand-crafted reward rules.

Termination Function. Each single dialogue, as one episode, will be terminated when both colour and shape knowledge are either taught by human tutors or known with high confidence scores.

Noting that, we applied the **SARSA algorithm** (Sutton and Barto, 1998) for learning the multi-MDP learning agent with each episode defined as a complete dialogue for an object. It was configured with a $\xi-$Greedy exploration rate of 0.2 and a discount factor of 1.

8.2 Experiment: Evaluation of the Optimised Agent on Overall Learning Performance

In this section, we designed an experiment to evaluate the RL-based optimised learning agent on the interactive learning task, where the agent will go through 500 visual instances from a hand-made visual object dataset (as introduced in Chapter 3).

8.2.1 Experiment Task & Procedure

This experiment involves: 1) whether the agent is able to handle natural, coherent conversations with the simulated user; and 2) whether the agent is able to effectively learn novel unseen visual concepts through interaction. As noted here, since spontaneous conversation with incremental dialogue phenomena (e.g. self -correction and -repetition) may lead to noise negatively impacting on the performance of language understanding and also learning accuracy. In this experiment, we employ a "well-behaved" simulated user, which is built upon the cleaned-up version of the BURCHAK corpus, which only contains clean conversations with massive variations. But without natural, incremental phenomena, for example, "this is a red square.", "what is the shape of this object?" and "Learner: a red square. Tutor: the colour is correct but shape is wrong,".

Similar to previous experiments in Chapter 7, for evaluating the performance of the system in each condition, we performed a 20-fold cross validation with 500 images for training and

100 for testing. For each training instance, the learning system interacts (only through dialogue) with the simulated tutor. Each interaction episode about an object ends either when both the shape and the colour of the object are discussed and agreed upon, or when the learner requests to be presented with the next image (this happens only in the Learner initiative conditions). Here, we define a learning step as comprised of 10 such episodes. At the end of each learning step, the system is tested using the test set. The values used for the Tutoring Cost and the Recognition Score at each learning step correspond to averages across a 20-fold validation (including those on the plots in Fig. 8.4a, 8.4b and 8.4c).

8.2.2 Baseline System

In order to further investigate the effects of the learned adaptive confidence threshold on the learning performance, we build a baseline system with hand-crafted learning rules. Following the previous work in Chapter 7, the system is a rule-based system built upon the investigated dialogue strategy: the agent/learner takes the *initiative* in dialogues, and takes into account *uncertainty* while interacting with the tutor. Instead of using synthetic dialogue examples, we rebuilt the rule-based system against realistic dialogues in the BURCHAK corpus. Here, for comparing the learned policy on more aspects of this rule-based system is assumed under three different conditions, as follows:

Condition 1: Constant Threshold. The first baseline system (blue curve in Fig. 8.4c) is the rule-based agent with a constant threshold, following the previous settings in Chapter 7, where the positive threshold is initialized as 0.95 and kept same through the learning process.

Condition 2 & 3: Hand-crafted Optimised Threshold. Both the second and the third systems (orange and grey curve in Fig. 8.4c) are also built with a hand-crafted adaptive threshold policy, in which the positive threshold is initialized as 0.95 and is incrementally decrease by 0.05 or 0.01 after each learning step (10 training instances).

i.e. different to the toy system built on synthetic dialogues in Chapter 7, the rule-based system under different conditions are all able to cope with multiple dialogue intents in a single turn.

8.2.3 Evaluation Metrics

Following the previous experiment setup in Chapter 7, we compare the optimised agent with baseline systems on the overall learning performance – considering both the cost to the tutor and the recognition performance of the learned meanings, i.e. the classifiers that

ground our colour and shape concepts. The research aims to find out the best policy that is able to maximise the trade-off between the learning accuracy as well as the cost by the tutor throughout the learning process.

In terms of the **Recognition Performance**, instead of using the recognition scores provided by Skočaj et al. (2009), we make use of the standard Accuracy to measure the proportion of correctness in the visual attribute classification at the end of each learning step. The accuracy will reflect how many visual attributes (colour and shape) are correctly identified and described in the learning period.

In terms of the **Dialogue/Tutoring Cost**, we keep applying similar settings as the previous experiment, but abandoning both costs of understanding and producing single words within conversations, because through interaction with real humans, the length of user utterances are unpredicted and out of control. Instead of the cost on the word level, we, therefore, take the cost of each dialogue turn into account in the new experiment. Meanwhile, in order to distinguish different dialogue actions, e.g. *"inform"* and simple *"acknowledgement/rejection"*, we slightly modify the cost of these actions as listed below:

Cost Type	Inform	Acknowledgement/Rejection	Correction
Value	5	0.5	5

TABLE 8.1: Table of Costs to the Tutor in Learning Process

Inform: provide each single visual concept to the learner/agent, *Acknowledgement/Rejection*: a simple confirmation, and rejection with further move, providing a single concept

8.3 Results & Discussion

This section presents the comparison results between the RL-based optimised learning agent and hand-crafted agents on the visual-attribute learning task. We will discuss these results on the learning accuracy, tutoring costs as well as their trade-offs.

8.3.1 Results

Table 8.3 shows example interactions between the learned RL agent and the simulated tutor on the learning task. The dialogue agent learned to take the initiative and constantly produces coherent conversations through the learning process.

Fig. 8.4a and 8.4b plot the progression of average Accuracy and (cumulative) Tutoring Cost for each of the 4 learning agents in our experiment, as the system interacts over time with the tutor about each of the 500 training instances.

Here, the vertical axes in these graphs are based on averages across the 20-fold validation - recall that for Accuracy the system was tested, in each fold, at every learning step, i.e. after every 10 training instances.

Fig. 8.4c, on the other hand, plots Accuracy against Tutoring Cost directly. Note that it is to be expected that the curves should not terminate in the same place on the x-axis since the different conditions incur different total costs for the tutor across the 500 training instances. this curve corresponds to the trade-off between accuracy and the cumulative tutoring cost in the learning process, which is the main evaluation measure we applied here to judge the system's overall performance in each condition. We also report statistical significance results for this measure, as below:

An independent T-Test shows that the optimised RL-based dialogue/learning policy has achieved significantly better performance in accuracy than the hand-crafted adaptive threshold policies ($p < 0.01, t = 1.640$ and $p < 0.01, t = 6.581$ respectively). The RL-based policy shows significantly less tutoring cost than the rule-based system with a constant threshold($p < 0.01, t = 7.987$). The yellow, RL curve is actually slightly better than the constant-threshold policy blue curve - discussed below. The average performance of different threshold conditions has been shown in the following table.

Threshold Type	**Accuracy**	**Tutoring Cost**	**Ratio**
Constant	0.8744	2304.57	0.000379
Adaptive-0.5	0.7505	677.16	0.001102
Adaptive-0.1	0.7837	1199.91	0.000653
RL-based	0.8573	1796.15	0.000477

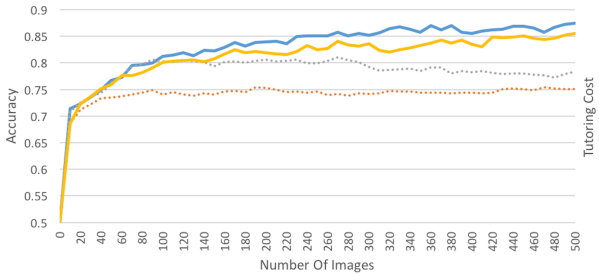
TABLE 8.2: Table of average performance of different Threshold Conditions

8.3.2 Discussion

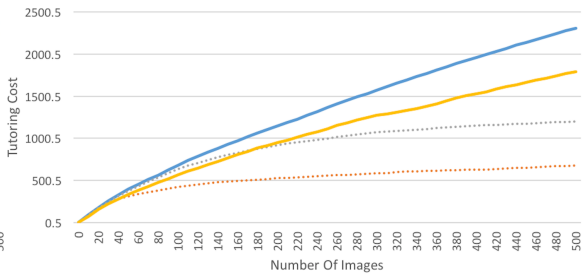
Accuracy As can be seen in Fig. 8.4a, the rule-based system with a constant threshold (0.95) shows the fastest increase in accuracy and finally reaches around 0.87 at the end of the learning process (i.e. after seeing 500 instances) – the blue curve. Both systems with a hand-crafted adaptive threshold, with an incremental decrease of 0.01 (grey curve) and 0.05 (orange curve), have shown an unexpected trend in accuracy across 500 instances, where the orange curve flattens out at about 0.76 after seeing only 50 instances, and the grey curve shows a good increase in the beginning but later drops down to about 0.77 after 150 instances. This is because the thresholds were decreased too fast, so that the agent cannot hear enough feedback (i.e. corrective attribute labels) from tutors to improve its predictions. In contrast to this, the optimised RL-based agent achieves much better accuracy (i.e. about 0.85) by the end of the experiment.

Dialogue Example (a)	Dialogue Example (b)
T: what is this object called? L: a red square? T: the shape is correct, but the colour is wrong. L: so what colour is this? T: green. L: okay, red.	L: blue? T: yes, blue is for the colour. and shape? L: sorry, i don't know the shape. T: the shape is circle. L: okay, got it.

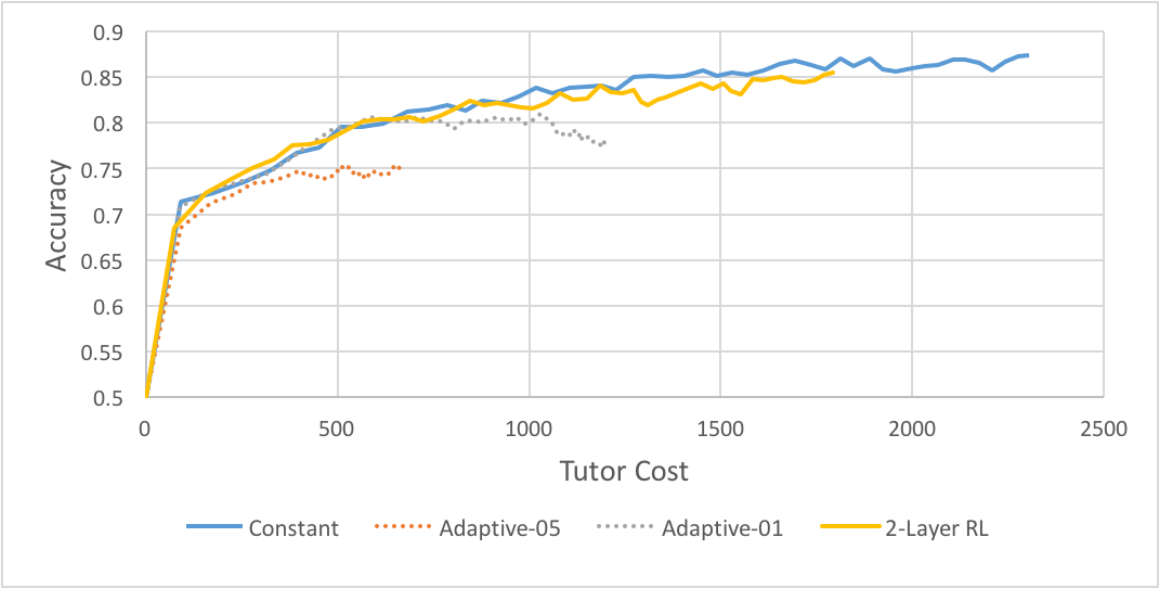
TABLE 8.3: Dialogue Examples between the RL-based Learning Agent and the Simulated Tutor: *(a) Tutor takes the initiative (b) Learner takes the initiative*



(A) Accuracy



(B) Tutoring Cost



(C) Overall Performance

FIGURE 8.4: Evolution of Learning Performance

Tutoring Cost. As mentioned above, there is a form of *active learning* taking place in the experiment: the agent can only hear feedback from the tutor if it is not confident enough about its own predictions. This also explains the slight decrease in the gradients of the curves (i.e. the cumulative cost for the tutor) (see Fig. 8.4b) as the agent is exposed to more and more training instances: its subjective confidence about its own predictions increases over time, and thus there is progressively less need for tutoring. In detail, the tutoring cost progresses much more slowly while the system was applying a hand-crafted adaptive

threshold (i.e. incrementally decreases by either 0.01 or 0.05 after each bin). This is still because the agent will not drive any conversations (i.e. no interaction) with the tutor, if the classification score is always higher than a certain positive threshold (for example, the agent has a threshold less than 0.65). In contrast, the RL-based agent shows a faster progress in the cumulative tutoring cost, but achieves higher accuracy.

Overall Performance. Here, we only takes into account the curves (the agent with a constant threshold, blue and the one with a RL-based threshold, yellow) that can achieve good recognition accuracy (over 0.8). Others with the incremental decreased threshold cannot achieve an acceptable learning performance. Fig. 8.4c shows that the agent with an adaptive threshold (yellow) achieves slightly better overall performance than the blue curve, as it achieves a comparable accuracy and request less effort/involvement from the tutor. We therefore conclude that the optimised learning agent, which finds a better trade-off between the learning accuracy and the tutoring cost, is more desirable.

8.4 Chapter Summary

This chapter is concerned with an interactive visual-attribute learning task, in which the agent is expected to optimise its behaviours to effectively learn unseen visual concepts (colour and shape) from the tutor incrementally, over time. We trained the initial system using Reinforcement Learning and a hierarchical MDP via dialogues with a simulated tutor, which is built on the cleaned-up version of the BURCHAK corpus. We evaluate this agent by comparing it with hand-crafted dialogue systems on the overall performance: a balance between recognition accuracy and cost by the tutor. This evaluation has shown that:

- One of the milestones of this research and also a challenge of this chapter is that an intelligent agent with an adaptive confidence threshold learns to determine whether or not it needs to request help from people. It is able to work on tasks by itself[2].
- The agent learns to retrieve the useful information (i.e. attribute words) from dialogues. It also learns to process natural, coherent conversations with the tutor, which is from the realistic data itself rather than a list of hand-crafted rules of the reward function.

[2]In the recent work, instead of acquiring visual concepts for toy objects (i.e. with simple colour and shape), we extend the learned optimised dialogue strategy to interactively learn about real object classes (e.g. `shampoo, apple`). The latest system integrates with a *Self-Organizing Incremental Neural Network* and a deep *Convolutional Neural Network* to learn object classes through interaction with humans incrementally, over time.

However, the initial optimised learning agent as proposed in this chapter, instead of learning from natural human-daily conversations, is trained only on the cleaned dialogues from the BURCHAK corpus, that covers massive variations but without natural, incremental dialogue phenomena. To our knowledge, some dialogue phenomena, such as self-correction, may bring more noise that negatively impacts both dialogue understanding and the overall learning performance on an interactive learning task.

In the next chapter, we aim to improve the learned dialogue strategy to understand/produce these natural, incremental conversations on learning by integrating the framework with the DS-TTR module. We will mainly focus on the ability of the new learning agent on handling self-correction phenomena in the learning process. Similar to this chapter, we will keep evaluating the new agent strategy by interacting with the simulated environment, and also we will compare the new policy with the existing one learned in this chapter to find out the best learning framework and conversation strategies for the visual concept learning task.

Chapter 9

An Optimised Learning Strategy on the Lexical Level for Interactive Grounding Tasks

The previous optimised learning agent presented above has shown good performance on achieving a balance between learning accuracy and tutoring cost through the interactive learning/grounding process, as well as processing natural, coherent conversations with the tutor simulation. However, different to what we initially planned (i.e. learning novel attribute knowledge though everyday incremental conversations), this dialogue policy from the previous chapter was trained and evaluated by interacting with a "well-behaved" tutor, which is built on the *cleaned-up* version of the BURCHAK corpus. Comparing to the original transcripts, the *cleaned-up* dataset lacks the interactional variations for natural, spontaneous incremental dialogue. In this chapter, we hypothesise that *a multi-modal framework incorporating incrementally constructed logical representations, such as DS-TTR (Eshghi et al., 2011), can show better performance on processing such incremental variations than one using dialogue act representations.*

In order to investigate this hypothesis, in this chapter, we extend the optimised learning agent by replacing the hand-crafted NLU component (i.e. SimpleSLU) with the DS-TTR model (Section 9.1), where we will attempt to cope with two essential challenges of the DS-TTR model: 1) grammar coverage for realistic data, and 2) mappings between DS semantic trees and Dialogue Acts (DA) (Stolcke et al., 2000). On the other hand, following the introduction of incremental processing with the DS-TTR model in section 9.1.2.1, compared to other phenomena, the"*self-repair*" phenomena is the only one that modifies the original semantic representation, which might lead to utterance misunderstanding, and so to worse visual attribute retrieval and even task failures. Hence, in this chapter, incremental

processing mainly focuses on coping with the "self-repair" phenomena in real-time conversation with humans or simulations. In section 9.2, we design a 2×2 factorial experiment to investigate the real impacts of the "*self-repair*" phenomena on the overall learning performance and also to assess the capacity of the newly proposed learning agent on incremental processing. Following the experiment, we will further explain and discuss the results on the aspects of progression of learning *Accuracy*, the cumulative *Tutoring Cost*, as well as their trade-offs (see Section 9.3). Eventually, all achievements in this chapter will be concluded in the Section 9.4.

9.1 Lexical-level Dialogue Agent for Learning

In this section, we introduce a new teachable system based on the previous version (in Chapter 8) that mainly concentrates on optimising the learning/dialogue strategies for learning novel visual attributes from humans incrementally, over time, whilst here we will focus more on its capacity for processing natural, spontaneous incremental conversations with human users. To achieve the goal, instead of employing a standard dialogue system with a hand-crafted NLU component, we extend the framework by interacting with a more robust model – the DS-TTR model. Although similar to the previous system that was trained on the cleaned up version of the real data, what we expect here is that the new agent can still handle the incremental phenomena while interacting with the tutor. More details about this solution are presented below:

9.1.1 Multi-modal Framework integrated with DS-TTR Dialogue Module

Following the introduction of the DS-TTR model in the Chapter 3, for processing everyday incremental conversations, we incorporate the general **Interactive Multi-modal Framework** with the DS-TTR dialogue model instead of the act-based, classical Natural Language Understanding (NLU) model, as detailed below:

9.1.2 DS-TTR Dialogue Module

The new multi-modal framework deploys an incremental word-by-word grammatical parser and generator – DyLan (Eshghi et al., 2011) – is built based on Dynamic Syntax (DS) formalism that incrementally construct a semantic tree by sequentially parsing each single word in an utterance. The DS model constructs the semantic tree by executing a set of computational and lexical actions. The contextual and semantic representations constructed

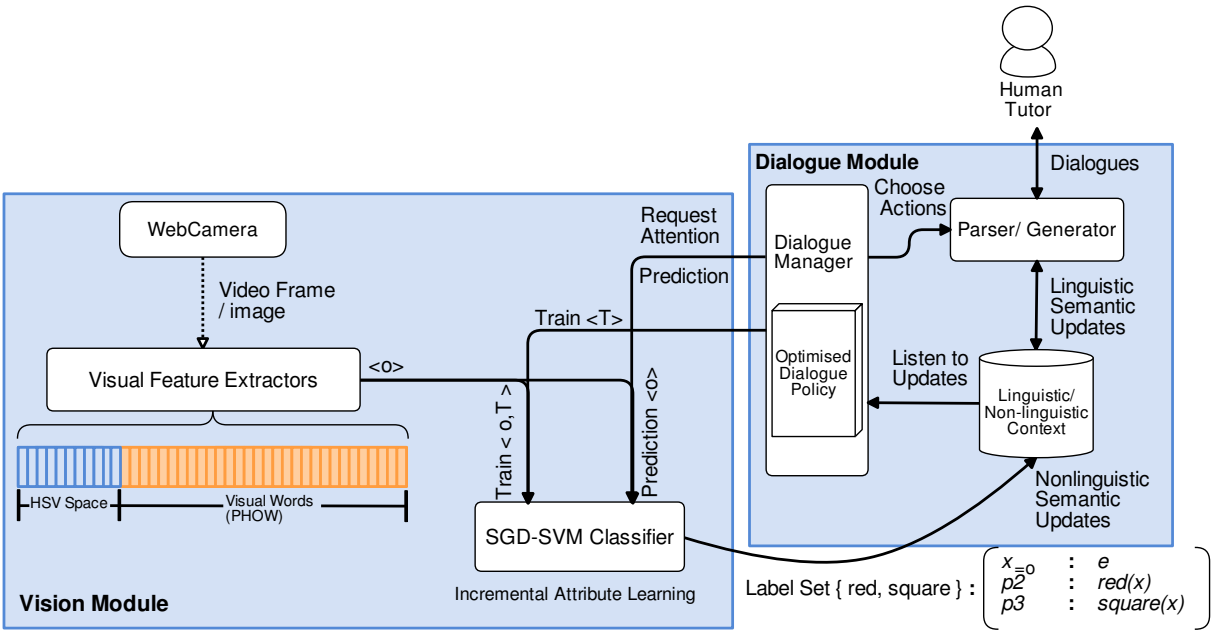


FIGURE 9.1: Multi-modal System Architecture integrated with the DS-TTR Dialogue Module

by DS are of the fine-grained semantic content that can be jointly negotiated or accepted by the speakers, as a result of processing questions, statements, corrections and etc. In recent work, Purver et al. (2011) incorporated DS with a logical formalism (i.e. Type Theory with records (TTR)), which is an extension of standard type theory that shows good performance on modelling not only dialogues, but also information from different modalities (e.g. vision and language) into a common semantic representation (see e.g. (Larsson, 2013)). Given the visual-concept learning task, we make the use of the DS-TTR model to model both linguistic (natural language conversations) and non-linguistic context (perceptual knowledge) in the form of TTR record types, in which the visual classification results are grounded as atomic items in the perceptual context. Hence, given a specific question (as part of the linguistic context), the agent is able to effectively retrieve the information from the perceptual TTR record types through unification, and vice versa.

Previous work by Eshghi et al. (2015), Hough and Purver (2014b), Purver et al. (2011) has demonstrated that DS-TTR model can show good performance on addressing natural, spontaneous conversations. In the following section, we will explain how DS-TTR addresses the incremental dialogue within the DAG visualization[1].

[1]Directed Acyclic Graph (DAG) is applied to visualise how the dialogue context and procedure can be characterised within the DS framework (see Appendix A)).

9.1.2.1 Incremental Processing with DS-TTR

Following the description from Chapter 5, many dialogue phenomena in the spontaneous interaction with human beings, e.g. *split utterances, hesitations, continuations, self -repetitions and -corrections*, have raised an incremental view of language processing. In this section, we will mainly explain how an incremental semantic construction framework (DS-TTR) can help with these phenomena in real-time conversations.

Regarding the protocol of the DS-TTR model, all contributions of these phenomena to the incremental semantic construction can be viewed as a type of repair on the semantic-level. According to the introduction by Eshghi et al. (2012), the repair that they are trying to address can be defined into two types: 1) a *repair* that involves a local partial restart of the reparandum, and 2) an *extension* that involves a local, further specification of the reparandum. To our knowledge, some dialogue phenomena, like *self-correction*, can be considered in the first type of repair (called type 1 below), and other phenomena (e.g. the *split utterance*, *hesitation*, *continuation* and *self-repetition*) can be considered as the extension one (called type 2). Here, we will briefly explain how these two types of repair can be addressed in the DS-TTR, as below:

- **Repair in Incremental Semantic Construction** Generally, the type 1 repair is invoked resulting from an on-line revision of the TTR goal concept, in which a new goal concept cannot be subsumed to the one that speaker had set previously (Eshghi et al., 2012). In other words, this repair will be applied when the DS-TTR model realises that there are not any chances to gain a succeeding word edge to extend the DAG while parsing/generating a new word. The repair procedure is applied to backtrack the context DAG (Eshghi et al., 2012) (see example in Fig. 9.2): restart the generation from the last word edge and then continually create a new DAG vertex (i.e. one vertex at a time) until the new partial tree subsumes the new goal concept. Once the subsumption has been achieved, generation will proceed as usual by "extending the DAG from that vertex". (Eshghi et al., 2012) explain that, instead of removing the the word edge backtracked, the model will mark it as "repaired", since that edge, as a public conversational record, should still be able to be re-accessed for later anaphoric reference or other purposes. Regarding particular dialogue phenomena, such as self -repair and -repetition, the main difference between these phenomena is that, the *self-repair* will modify the record type goal concept with *new* concept values, which usually don't occur in other phenomena, where the new goal concept is still a sub-type of the previous one. Hence, instead of backtracking along the incrementally available context DAG with a new vertex, processing other phenomena, such as *continuation*,

self-repetition, can simply extend the context DAG, similar to the type 2 of repair (see below).

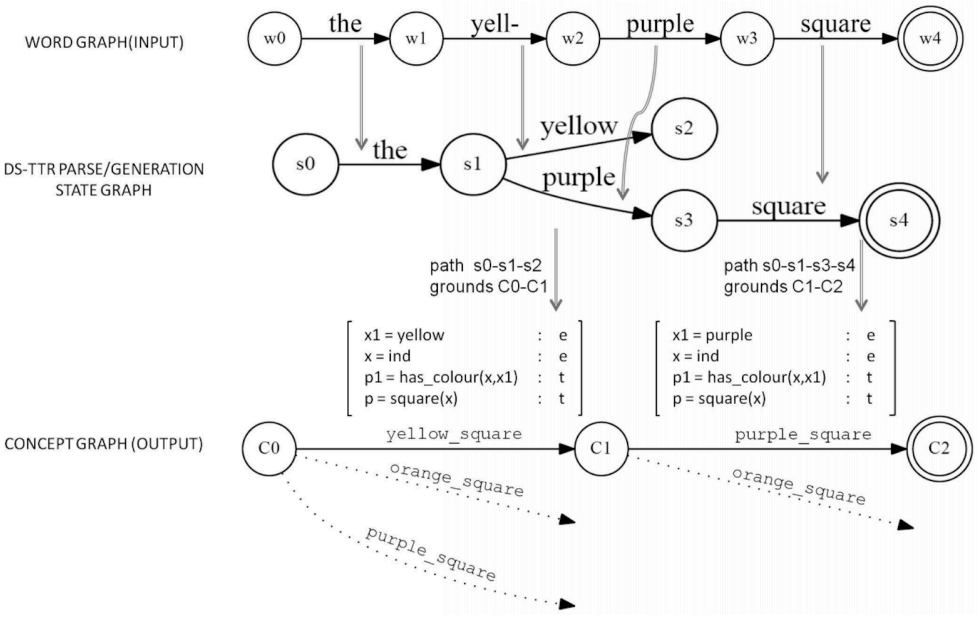


FIGURE 9.2: Incremental DS-TTR Parsing of a self-repair for utterance "the yell-(ow) purple square"(Hough and Purver, 2012)

- **Extension in Incremental Semantic Construction** On the other hand, the type 2 repair (i.e. *extension*) is usually invoked in transition relevance positions in the conversation after complete or incomplete turns. The DS-TTR parser considers this type of repair as a simple extension of the matrix tree (see examples in (Cann et al., 2005a)), which leads to an extension of the subtype of the root TTR record type. In the case of parsing/generating "I go to Paris · from London", the change in the goal concept will not invoke the backtracking procedure.

Backtracking is only invoked at a semantic-syntax mismatch, in which the revised goal concept cannot be longer subsumed to the root record type for utterances/dialogues so far realised (Eshghi et al., 2012).

In this thesis, this is the first time that the DyLan parser has been used to process real conversations between humans. For dealing with realistic conversations between humans, with lots of variations and uncertainty, we need to improve the DS-TTR lexicon to cover as many variations and uncertainties as possible in the real interaction (see more details in the next section).

On the other hand, we loosely follow the theory of the lexicon-based incremental dialogue management from Kalatzis et al. (2016b), which, avoiding dialogue act representations, generates a new semantic goal concept (TTR record types) given a set of decomposed

semantic features. In this thesis, we incrementally parse each utterance word-by-word, and then infer a dialogue act tag for the semantic tree at the end of parsing process following a set of learned grammar rules (more details are presented in Section 9.1.2.3)

9.1.2.2 Lexicon & Grammar Coverage

Similar to other semantic parsers and generators, enriching the lexicon and creating grammars to handle realistic dialogues is obviously a major task and also one of the biggest challenges for the DS-TTR model in support to implement a robust, practical dialogue system. In the thesis, given the learning domain, we have eventually managed to cover around 96% of real data [2] from the BURCHAK corpus (Chapter 5) and also a variety of utterance-pairs between an interactive system and the user simulation. One of the most essential but common features we did not cover in the parser is spelling and grammatical mistakes. Although these mistakes were frequently occurring in real data, we will not take them into account for the enrichment of the lexicon grammar, due to the fact that: 1) satisfying these features/mistakes might lead to an extremely flexible DS-TTR model that infinitely increases the searching/parsing complexity for both parsing and generation process and loses the meanings of introducing the relevant grammars; and 2) in a real speech-based system, the ASR will automatically filter out all spelling mistakes/typos. Here, all these mistakes will be filtered out only from the cleaned up version of the BURCHAK corpus, which is applied to train the user simulation and the learning agent in this chapter. Noting that for keeping the DS-TTR module being transferable across different domains, we attempted to make the DS-TTR grammar as generalised as possible in this implementation, although its generality will not be demonstrated in this thesis.

9.1.2.3 Dialogue Act Inference

In previous work, Purver et al. (2011) proposed a novel functionality that supports the DS-TTR model to map TTR record types to particular domain concept frames (for example Dialogue Acts) using "a simple XML matching specification". For the same reason as concepts in Purver et al. (2011), since TTR record types give fine-grained semantics that contain too much information and are difficult to learn over/from, more abstract representations, e.g. Dialogue Acts (DA), are needed to represent the content relevant in a particular domain, for example, visual-attribute learning task in this thesis. Therefore, here, we introduce a Dialogue Act inference mechanism, in which DA is derived using specialised rules

[2] The clean-up version of the BURCHAK corpus contains 1553 utterances in total. The new DyLan model, with more lexicon and grammars, can successfully parse 1490 utterances with correct TTR semantic trees and representations.

which take the same form as Computational Actions in Dynamic Syntax Grammars, i.e. IF-THEN-ELSE (see examples in Figs. A.2 and A.3 in Appendix A). We call these rules Dialogue Act Grammar. In general, Dialogue Act Grammar is applied to map DS trees (including Maximal Semantics RT) to specific DAs.

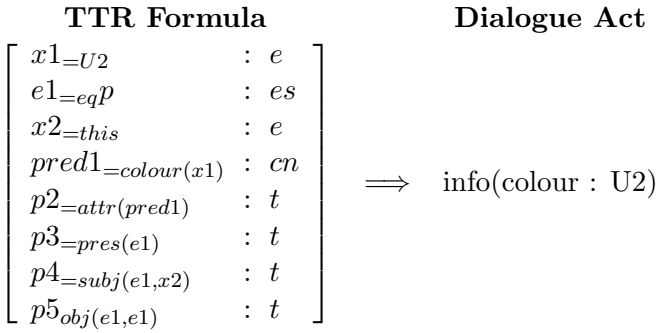
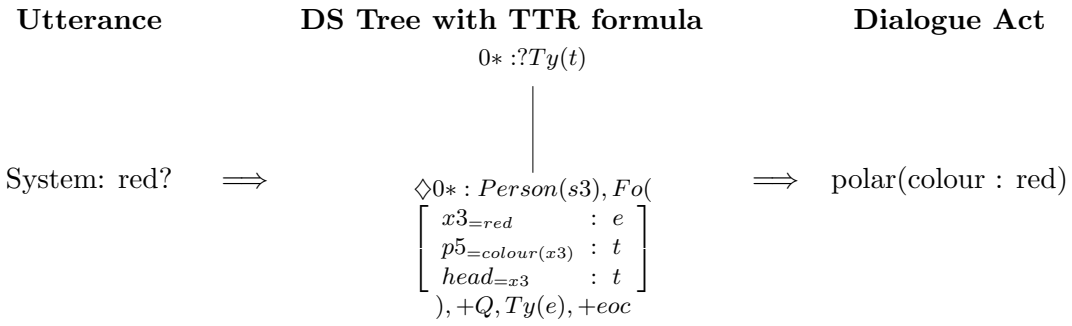
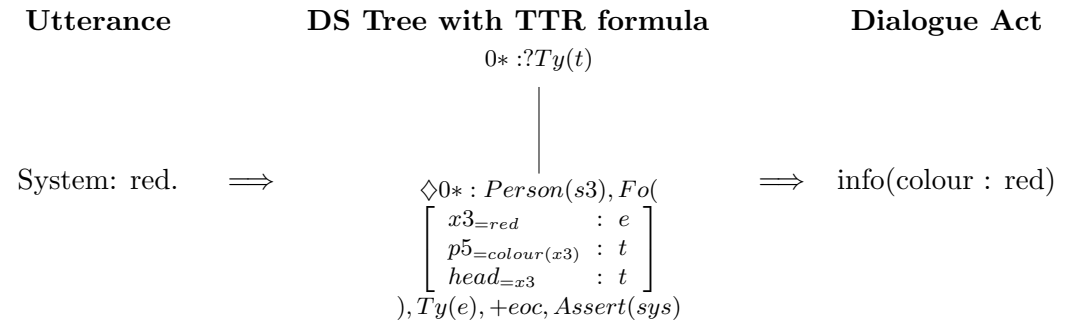


FIGURE 9.3: Example of mapping TTR Record Type to DA

More specifically, this DA derivation is done via inference over TTR Record Types (see example in the Fig.9.3). However, as inference with Record Types is not always sufficient to distinguish different dialogue acts, we also take into account other features of the actual DS tree, for example, dialogue acts, e.g. "*polar(colour)*" and "*info(colour)*", are referred to the same TTR record type but with different DS trees (with different features). The way of distinguishing these two actions is to check whether the root node of the DS tree contains a feature of "+Q" or not (see Fig. 9.4).



(a) mapping DS tree to dialogue act "polar(colour)"



(b) mapping DS tree to dialogue act "info(colour)"

FIGURE 9.4: Example of mapping different DS trees (with same TTR record types) to DAs

The first 5 or 6 lines of most Dialogue Act grammars here are applied to check whether the current state is on a type-complete sub-tree. The current version of DA inference is *not done strictly incrementally*[3] in this thesis, but only at points where there is a complete semantic formula of a particular type, e.g. proposition (t) or entity (e). In the former case, the model will check whether there is a complete sub-tree of type t, and in the latter case type e. In the other words, the DAs can only be inferred after a completed utterance within dialogue, (see an example from BURCHAK corpus in Fig. 9.6)

Learning Dialogue Act Grammars for Inference Here, we will briefly describe how the DA grammars are learned/collected automatically from the annotated data (the BURCHAK corpus in Chapter 5) (see the *pseudo-code* in Appendix C). We note that all dialogues in the corpus have been fully annotated with dialogue-act representations (see Table 9.1), where dialogue turns will be separated into multiple turns based on their action annotations, each turn can have a single utterance and only be assigned with a unique dialogue act.

Utterance	Annotation
usr: what colour is this object?	*ask(colour)*
sys: red.	*info(colour=red)*
usr: good job.	*accept(colour)*
usr: and shape?	*ask(shape)*
sys: is this square?	*polar(shape=square)*
usr: no.	*reject()*
usr: it is a circle.	*info(shape=circle)*
sys: okay.	*accept()*

TABLE 9.1: Dialogue Example with annotations from the BURCHAK corpus.

Before learning different DA grammars, we predefine a number of grammar templates, which contain all features/conditions of a completed DS sub-tree for distinguish different Dialogue Acts. Each action may be associated with one or more templates to satisfy different semantic conditions in the parsing states. This learning process begins with parsing each utterance within annotated dialogues, getting a completed DS sub-tree with a particular TTR (see lines 5 to 13 of the pseudo-code in Appendix C). The model attempt to check whether the semantic tree can be correctly tagged by one of the existing DA inference grammars (lines 14 to 19). If not, it will get all possible grammar templates mentioned above for the certain action (e.g. 'info(colour)', 'ask(shape)' and etc.), otherwise, jump to the next utterance directly (line 15). The model will select a unique grammar template, in which the semantic tree can completely fulfil all required features in the *IF*-conditions (lines 21 to 24). Meanwhile, given the selected grammar template, the model will also double check whether there has been an inference grammar learned previously that deploys the same

[3] As part of the future work, this is essential for a fully incremental DA inference, i.e. instead of waiting for after completed sub-trees, DS model should be able to infer the DAs after parsing each single word through real-time conversation (see more discussion in Chapter 10).

grammar structure with same TTR record type (lines 26 to 30), if not, a new DA inference grammar can be created using the selected grammar template with a Maximal TTR record type on the pointed node of the semantic tree (lines 31 to 33). Note that, all atomic items (or specific visual-concept labels) in both formula and inference grammar will be abstracted with a Meta variable (line 31), for instance "colour : $P8$" and "shape : $U9$" (see example of learning DA grammar in Fig. 9.5).

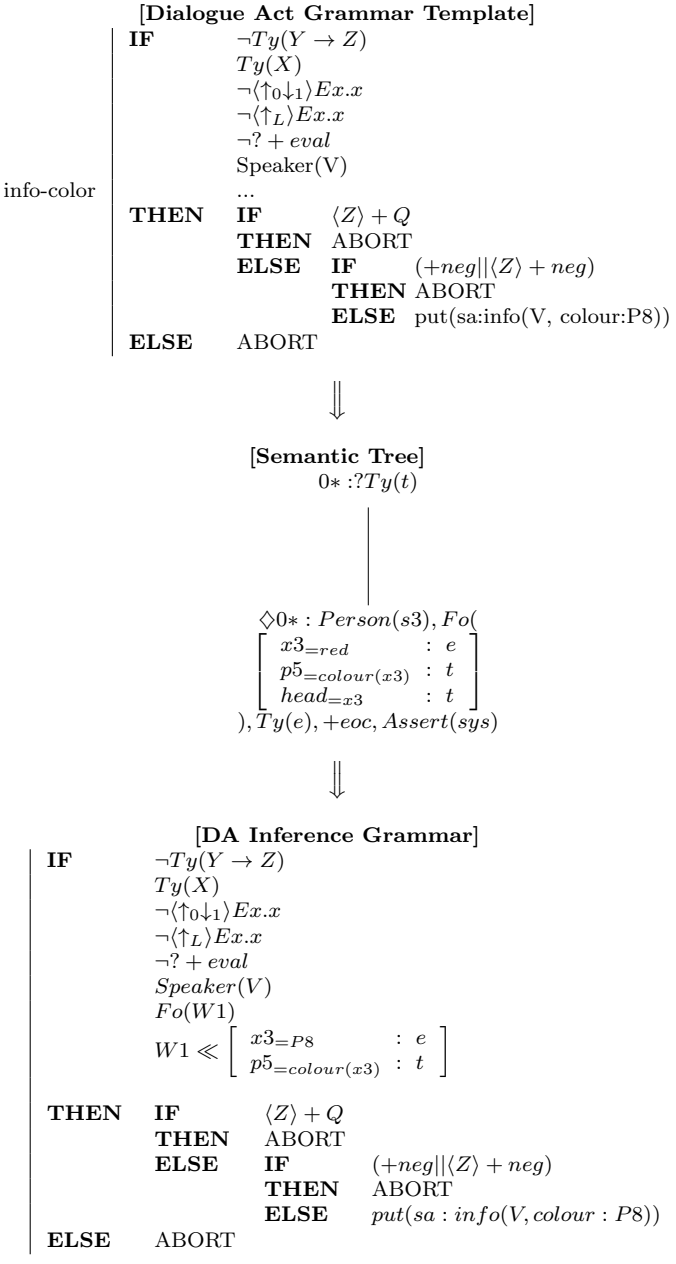


FIGURE 9.5: Learning a new DA Inference Grammar Rule using the template (after parsing the utterance:"sys: red.")

Generalisation over Inference Rules Through the learning process, there are many DA rules/grammars collected, in which many conditions in common. However, we do not need to have one rule per processed utterance. In order to reduce the search space for inference,

the model is designed to avoid any duplications, i.e. before adding new rules, the model will try to infer the DA of a specific utterance by iteratively executing all existing grammars under the same action (see lines 17 to 19 of the pseudo-code in Appendix C), if successful, ignoring the new rule, otherwise adding it into the DA grammar list.

At the end of the learning stage, all learned DA grammar rules will be automatically ordered by their formula specificity: the DS-TTR model will sort all learned DA grammars from the most specific to the most general comparing their conditional record types. The subtype relation in TTR is a partial order, which determines a natural partial ordering for the rules that involve record types. This means that searching for the correct DA rule is equivalent to lattice search (see Eshghi et al. (2013)), and is therefore fast: $O(\log n)$, though this is not yet fully implemented.

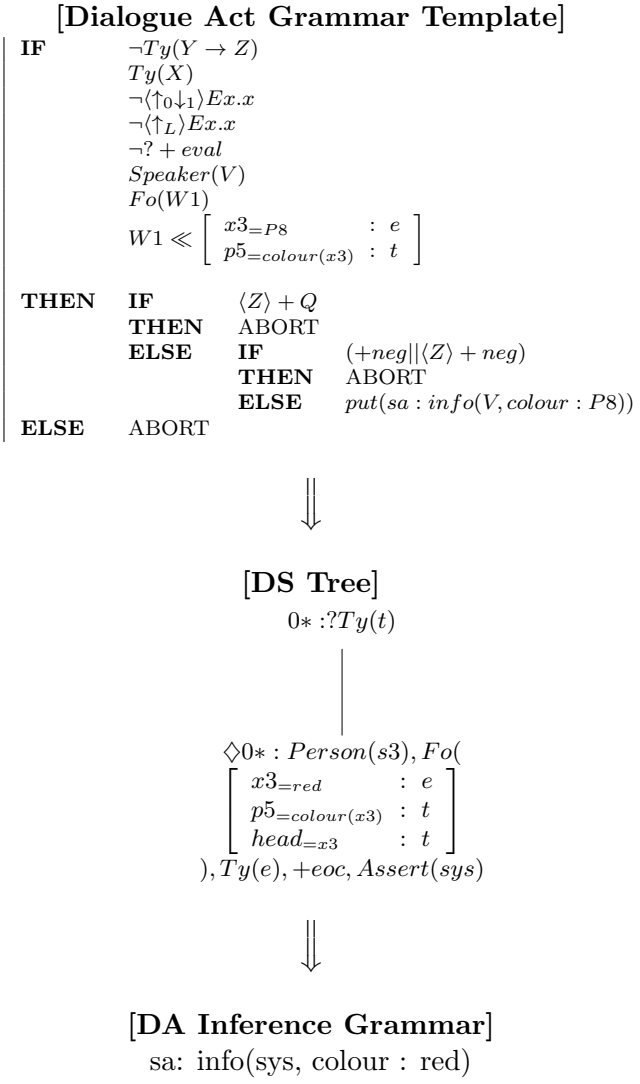


FIGURE 9.6: DA Inference through real-time conversation "sys: red. usr: good job."

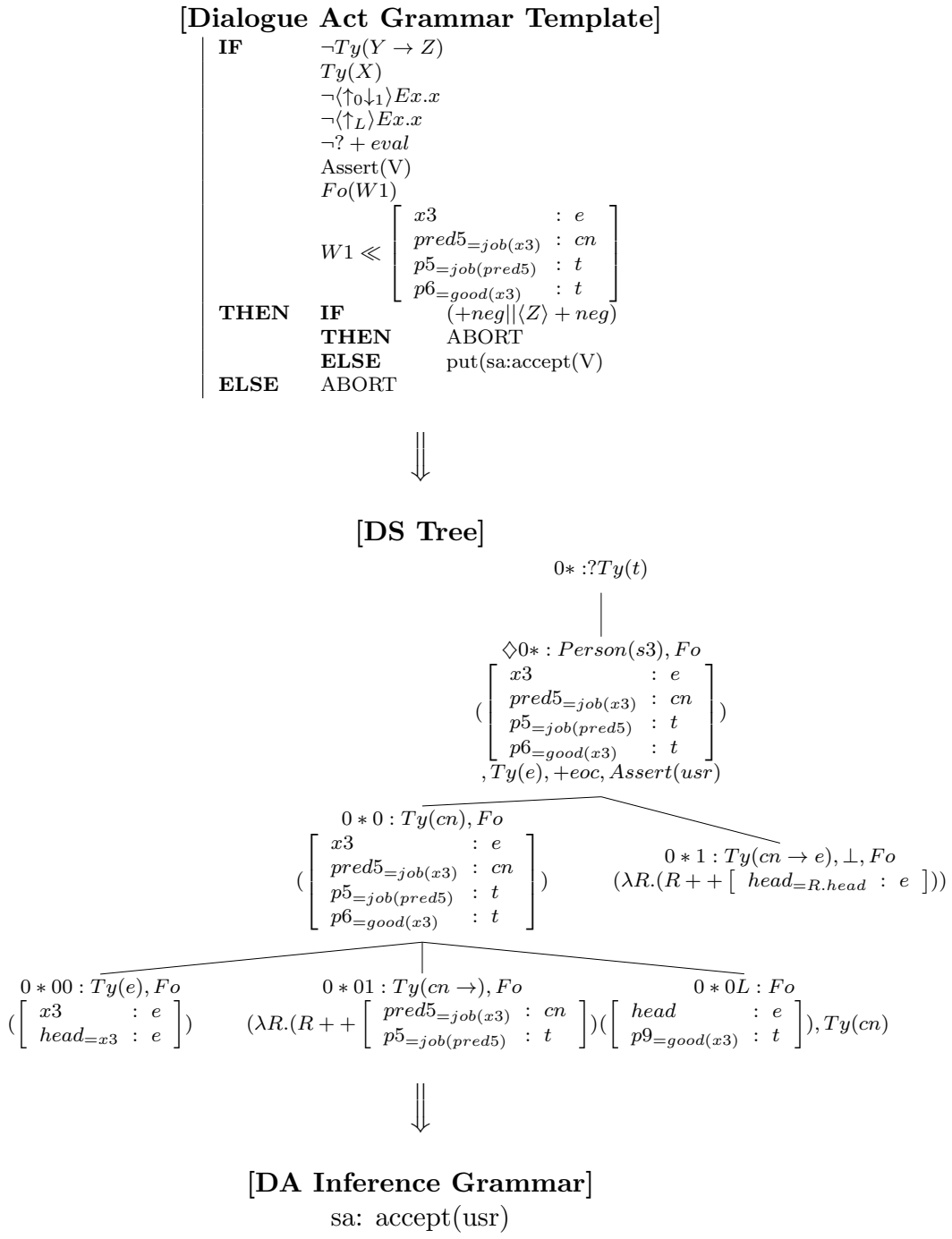


FIGURE 9.6: DA Inference through real-time conversation "sys: red. usr: good job."

9.2 Experiment Setup

In this section, we present a factorial experiment for evaluating the newly proposed RL-based learning policy while interacting with the tutor through an interactive learning process. Two main purposes of this experiment are to 1) show that without a proper processing of the incremental phenomena (i.e. "self-repair") might lead to misunderstanding of the

tutor utterance and so worse overall performance in such learning task; and 2) evaluate the capability of the new learning agent on processing such "self-repair" phenomena through real-time interaction with users by comparing it with the previous system that integrated an act-based hand-crafted NLU component. Regarding the details of this experiment, we will partially follow the experiment configuration from the previous chapter (see Chapter 8), including the evaluation metrics and cross-validations as well as the visual object dataset applied to learn the visual attributes.

As described in the previous chapter, the agent learns to perform a form of *active learning*, i.e. only asking for human feedback when it does not believe its own visual classification based on their confidence scores (see Chapter 8). It means that both the "active learning" behaviour and misunderstanding of the "self-repair" phenomena in dialogue might lead to the same negative effects in the learning/grounding task: learning visual instances with incorrect attribute labels. Therefore, instead of applying an adaptive threshold like in the previous chapter, we deploy a constant confidence threshold (0.95), where the system has much less chance to perform such *"active learning"*. This limits the conflict between self-repair and active-learning in our investigation and analysis of the capability incremental dialogue processing.

9.2.1 Design

In order to assess the capability of the new optimised learning agent on incremental processing, we carry out an experiment with a 2×2 factorial design, where each factor has two levels: (1) *system-type* (*Simple*/*DS-TTR*): represents different system versions, either the previous RL-based system integrated with a typical dialogue system and a simple hand-crafted Dialogue Act Tagging model (SimpleSLU) that converts the user utterance into a specific dialogue act via a pattern-matching method, or with the DyLan parser (described above) partially on the lexical level; and (2) *self-repair* (*+RE*/*-RE*) determines whether the simulated tutor simulates the "self-repair" dialogue phenomena while interacting with the learner/agent (e.g. "the colour ... no ... the shape is a square." or "this is a red ... wait ... a green triangle."). As mentioned above, the 'self-repair" will determine the learner's overall performance: trade-off between the learning accuracy and the dialogue cost from the tutor.

9.2.2 Incremental User Simulation for Grounding Task

To run this experiment for evaluating the capability of incremental processing, we built an "*incremental*" simulated user based on natural real-life dialogues from the BURCHAK corpus (Chapter 5), which is able to not only resemble humans' behaviours in the learning

task, but also model the disfluencies in everyday interaction in real-world environment by producing natural, incremental dialogue phenomena (*self-repair*), as presented below:

- ***self-repair***, e.g. "this is a red uhm sorry green square", will impact the task-specific information. In contrast to the other phenomena (e.g. self-repetition, continuation and fillers), this phenomena will lead to a modification of the representation with different label-values, which might affect the understanding of the user utterance, user experience, and even the task success.

Since the BURCHAK corpus is collected via a character-by-character incremental dialogue experimental toolkit (DiET) (Crocker et al., 2000, Ferreira, 1996, Purver et al., 2009b), it is hard to present a distributional percentage of the particular phenomena (*self-repair*) from the original transcriptions, where the turn is difficult to define. Hence, here, our phenomena simulation, with respect to the self-correction in particular, is as observed in other real-world data: 40% in a corpus of human-human goal-oriented dialogues, called the Map Task (Anderson et al., 1991). The simulation mode will generate the "self-repair" dialogue phenomena while interacting with the optimised learning agent (see Chapter 6 for how the phenomena is produced). All *self-repair* phenomena in this simulation model will be produced all within a single user utterance, rather than across multiple turns (see dialogue examples in Fig. 9.2). This setting can be turned on/off in the simulated user model as the binary factor (+RE/-RE). Both RL-based learning agents (mentioned above) are trained only with the non-incremental (-RE) simulation.

9.3 Results & Discussion

The section presents and discusses the comparison results between different factorial conditions, for the act- or lexical-level RL-based agent interacting with a user simulation with or without the "self-repair" phenomena generation, given the interactive visual-attribute learning tasks. In the results, we take into account the learning accuracy and tutoring costs as well as their trade-offs.

9.3.1 Results

Some example dialogues between the new optimised learning agent and the simulated tutor on the learning task are shown in Table 9.2. Comparing to the previous learning agent, the new one has learned to process natural, coherent and incremental conversations during the learning period.

Dialogue Example (a)	**Dialogue Example (b)**
T: so what is this object?	
L: i don't know.	L: so is it a green triangle?
T: it is a red wait blue square.	T: close. the colour sorry the shape is circle.
L: okay.	
T: can you repeat again?	L: okay and colour?
L: blue square?	T; it is green.
T: yes, well done.	

TABLE 9.2: Dialogue Examples between the RL-based Learning Agent and an incremental Simulated Tutor (i.e. generating incremental phenomena of "self-repair") : *(a) Tutor takes the initiative (b) Learner takes the initiative*

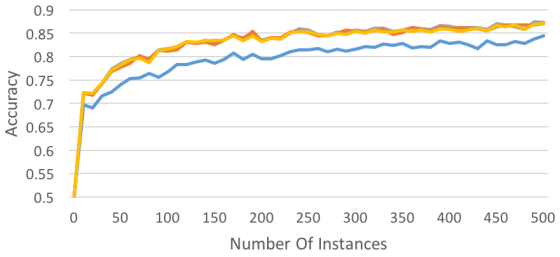
On the other hand, Figure 9.7a and 9.7b present the progression of the average *Accuracy* and cumulative *Tutoring Cost* respectively through the interactive learning process with 500 training instances in our experiment. As noted, the vertical axes in these graphs are based on averages across the 20 folds - recall that for Accuracy the system was tested, in each fold, at every learning step, i.e. after every 10 training instances

In addition, Figure 9.7c plots the curves of average *Accuracy* against *Tutoring Cost* through the learning process. Different to the former two figures, here the curves are not expected to terminate in the same place on the x-axis, because different factorial combinations may lead to different cumulative costs towards the end of the learning process. The gradients of these curves are assigned to the trade-off between the progressive Accuracy and the cumulative Cost: *increase in Accuracy per unit of the Tutoring Cost*. It constitutes the overall performance of each system under specific conditions. Here, we report statistical significance results for both the progressive *Accuracy* and the cumulative *Cost*, as below:

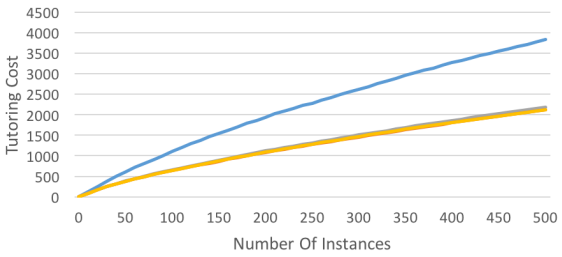
The Paired Samples T-Test shown that the new agent associated with the DS-TTR model significantly outperforms the SimpleSLU-based one in the learning task through natural, incremental dialogues. More specifically, the SimpleSLU-based agent show a significant average difference on both accuracy ($t_{19} = -5.424, p < 0.005$) and cost ($t_{19} = 61.970, p < 0.005$) between through incremental (the blue curve) and non-incremental conversations (the grey curve). The new agent (orange curves in Fig. 9.7c) shows significantly higher accuracy ($t_{19} = -4.060, p < 0.005$) but less cost ($t_{19} = 57.576, p < 0.001$) than the SimpleSLU-based one (blue curves) while interacting with an incremental (+RE) simulator. The mean gradients of all other curves factorial conditions are better than the condition of the SimpleSLU-based agent with the incremental simulated tutor. The average performance of different Systems, within incremental and non-incremental conversations, has been shown in the following table.

Learning Agents	Accuracy	Tutoring Cost	Ratio
SimpleSLU-RE	0.8738	2186.05	0.000399
SimpleSLU+RE	0.8448	3832.80	0.000221
DS-TTR-RE	0.8707	2120.33	0.000411
DS-TTR+RE	0.8705	2118.53	0.000411

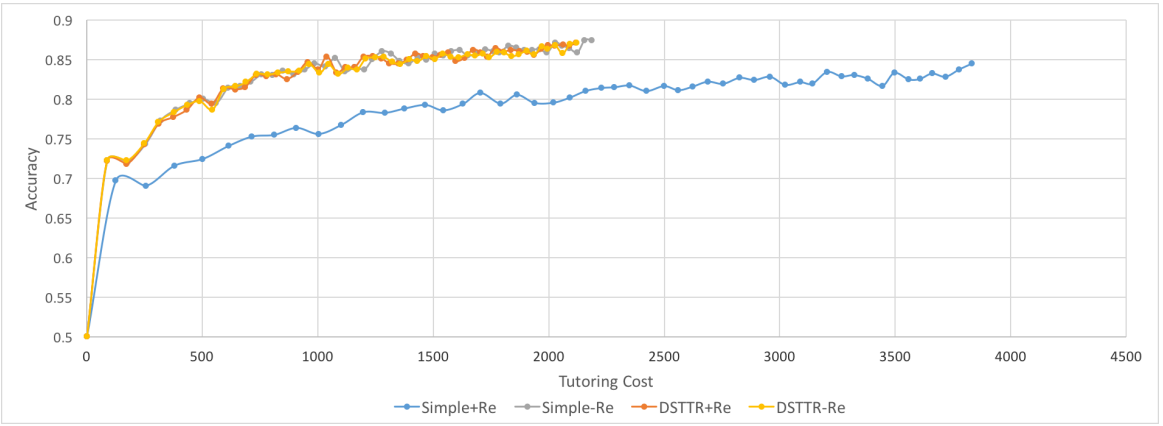
TABLE 9.3: Table of average performance of different Systems within incremental and non-incremental conversations



(A) Accuracy



(B) Tutoring Cost



(C) Overall Performance

FIGURE 9.7: Evolution of Learning Performance

9.3.2 Discussion

Accuracy Fig. 9.7a indicates that, the SimpleSLU-based learning agent shows the slowest increase in accuracy and almost flattens out at only 0.80 (the blue curve) while interacting with an incremental simulated tutor (+RE). This might be because the hand-crafted DA tagging model is unable to handle the "self-repair" dialogue phenomena in real conversations, i.e. it easily misunderstands what information the tutor truly wants to tell and also updates the classifier with wrong attribute labels, which lead to worse recognition performance towards the end of learning process. It also indirectly proves that the "self-repair" phenomena does affect the success of the learning task. In contrast to this factorial combination (Simple+RE), the agent using the DS-TTR model shows consistently good performance

when it learns novel visual knowledge by communicating with both non-incremental and incremental simulations, both of which achieve about 0.88 and 0.87 of the learning accuracy respectively.

Tutoring cost Similar to the accuracy results, the SimpleSLU agent presents a significantly higher tutoring cost than the other conditions, costing around 4000 units, which is nearly twice as much as that other agents did through the learning period. For the same reason that such agent cannot correctly understand user utterances in incremental dialogues, it leads to more corrections/attribute-label statements (i.e. which induces a higher cost than acceptance (see the evaluation metrics from Chapter 8)) by the tutor than the other conditions. By contrast, the DS-TTR learning agent can constantly presents comparable touring cost (2120 units), no matter interacting with a "well-behaved" or incremental tutor simulation.

Overall Performance Since we are mainly concerned about the capability of the learning agent on processing natural, spontaneous dialogue given the interactive grounding task, we only compare the gradients of the curves between the DS-TTR-based agent (orange curve) and the hand-crafted SimpleSLU agent (blue curve) while interacting with an incremental (+RE) simulator in Fig. 9.7c. The DS-TTR agent achieves significantly better overall performance than the SimpleSLU one, i.e. it achieves the same accuracy but with much less dialogue effort/cost. Therefore, we conclude that such learning agent, incorporating with DS-TTR framework, is more desirable given the task of incrementally learning visually-grounded word meaning through daily, incremental conversations with human beings.

9.4 Chapter Summary

In this chapter, we attempt to address a gap in the literature that learning visually-grounded word meanings through *everyday*, *incremental* conversations with real humans, instead of "well-behaved" utterances/descriptions. We proposed a new optimised learning/grounding agent that incorporates with a logistic formalism parser/generator (DS-TTR model (Eshghi et al., 2012, Purver et al., 2011)) that can incrementally construct the semantic tree for modelling specific dialogues and multi-modal information (for instance, vision and language). In this case, we loosely follow the theory of lexical-level dialogue management from Kalatzis et al. (2016b) to incrementally process the utterance word-by-word, but instead of making a list of decomposed semantic features as MDP features, we map the completed semantic sub-tree to specific dialogue acts via a novel DA inference approach. Through the final experiment by comparing the newly proposed agent with the previous version (one with a hand-crafted pattern-matching NLU model (SimpleSLU)), we can conclude that this chapter makes two essential contributions on:

- showing that the incremental dialogue phenomena ('self-repair') does negatively affect the utterance understanding of the system and also the task success given the interactive learning task, if the system is unable to properly cope with these.
- we achieved an optimised learning agent that can still achieve better trade-offs between the learning performance and conversational tutoring cost through natural, incremental conversations.

Noting that, this is the first time that the DyLan incremental parser is deployed to process realistic human-human conversations.

From here, we have completed a list of research work in support to implement an optimised and interactive learning agent for addressing the visual language grounding problem.

In the next chapter, we will conclude the work presented for the interactively visual language grounding task in this thesis, as well as discuss its correlations with 4 core research questions raised in the beginning of this thesis. We will summarise the main contributions of this thesis, but also discuss the relevant limitations in this work. Finally, we will overview some potential research directions/work that might be carried out in the future.

Chapter 10

Conclusion & Future Work

In this thesis, we aim ultimately at addressing an interactive language grounding problem, i.e. learning to align symbols in a language (words, phrases and sentences) with aspects of the physical environment (e.g. attributes of objects) through Natural Language interaction with humans. We began this thesis by introducing the symbol grounding problem, as well as a number of previous works which have achieved great progress on the grounding problem via a diverse set of techniques, architectures, and models in a variety of tasks. Through this review, we noticed that language grounding cannot be effectively resolved until a list of essential properties are well taken care of, for instance, the system/robot should learn groundings compositionally, via Natural Language conversations, following an optimised dialogue policy, and so on. However, despite their importance, the previous work only takes into account one or some of these properties in their approaches.

We designed and implemented an appropriate solution with this in mind, noting that, to our knowledge, such an approach/system, taking into account all grounding properties at once, has not yet been accomplished. Here, we proposed a modular ***Interactive Multi-modal Framework*** in support of building a teachable robot/interface in this thesis. The framework has several desirable properties (see Table 10.1), i.e. it is *compositional*, *optimised*, trainable *incrementally* with *small amounts of data*, and able to handle *natural, spontaneous dialogue*.

This work	Compositional	Interactive (NL Conversation)	Optimised	Attribute Grounding	Natural Language	Incremental Learning	Semantic Representaion
This work	✓	✓	✓	✓	✓	✓	✓

TABLE 10.1: The work in this thesis addresses several desirable properties for interactive language grounding

More specifically, the framework mainly consists of two modules, i.e. a vision module in charge of constructing the visual, non-linguistic context in dialogue based on the visual classification output, and a dialogue module incorporating an incremental semantic parser

(DyLan (Eshghi et al., 2011)) and an optimised reinforcement learning based dialogue management, which is in charge of processing natural, everyday human conversations about the physical environment. In this thesis, we explored an incremental visual classifier (SGD-SVM model (Zhang, 2004)), which fits into the interactive learning/grounding task, i.e. incrementally learns low-level visual features, starting with no training examples, through real-time conversations. The SGD-SVM model has also been shown to be able to learn attributes from every single instance faster than other machine learning approaches. In this work, we applied the SGD-SVM model to train visual attributes as a set of binary classifiers (e.g. redness or not). Thus the system, from the visual perspective, treats all attributes equally without distinguishing their ontologies/categories.

We then turned to investigate and replicate human dialogue behaviours in an interactive learning task in Chapter 5 and 6. We collected a number of realistic human-human conversations for interactively learning visually-grounded word meanings (in the invented visual-attribute language) through ostensive definition by a tutor to a learner. To our surprise, the corpus contains not only a variety of dialogue capabilities and strategies in such a simple domain (i.e. only talking about colours and shapes of visual objects), but also a wide range of linguistic phenomena as encountered in natural, spontaneous dialogue (for instance, self -repairs and -repetitions, continuations, overlaps and fillers), which is one of the biggest challenges but also essential milestones of this thesis. For addressing those conversation issues in the grounding task, we presented a generic n-gram user framework for building a simulated tutor that resembles all dialogue capabilities, strategies and corresponding expressions, as well as simulates the incremental phenomena (or called speech disfluencies). We hypothesized that those dialogue capabilities, strategies, and even phenomena (specifically self-repair) are likely to impact on the learning/grounding performance of the agent.

Given the experiments in Chapter 7, we can conclude that, in order to approach good overall learning/grounding performance, i.e. achieve better learning accuracy but with less dialogue effort from the tutor, an agent should be able to: 1) take the initiative in conversations, 2) properly handle uncertainty from visual classification, 3) process context-dependencies from dialogue, and 4) demand further information if necessary. These capabilities of an agent should be automatically learned from realistic human data.

Hence, in Chapter 8 and 9, we then build our first learning/grounding system with optimised dialogue strategies using Reinforcement Learning and a two-layer multi-objective MDP. The strategy with all investigated capabilities and strategies has learned when to learn from the tutor. i.e. the agent only asks for further help from the tutor if necessary, as well as how to interact with humans, i.e. managing a natural, human-like conversation. In parallel, we incorporated and extend the incremental semantic parser (DyLan (Eshghi et al., 2011) based on the DS-TTR semantic formalism) with the proposed framework for

processing incremental, spontaneous dialogue (specifically self-repair), to understand what the meaning of symbols (visual-attribute words, e.g. "red" and "square") by back-tracking the previous context within a more complicated, noisy learning conversation (see more details in Chapter 3 and Appendix A). The DS-TTR formalism presents compositional semantic representations for both linguistic and non-linguistic context. As another milestone of this thesis, we introduce a mechanism of dialogue act inference that predicts the most appropriate dialogue act based on completed semantic sub-trees within the DS-TTR framework following a set of Dialogue-act inference grammars in an IF-THEN-ELSE structure. Since these rules, automatically learned from the real data, are generalised, they are transferable across domains and contexts. We also note that this is the first time that DyLan (Eshghi et al., 2011) is implemented for realistic human dialogues.

10.1 Discussion

In this section, we attempt to discuss one of the likely limitations that we have noticed but did not address in this thesis: Learning *visual classifiers* or *vision-language mappings*. We will also explain why they cannot be coped with properly at this moment. Following all discussions (including the one discussed below) we previously mentioned in this thesis, we raise several new challenges for the current framework, and we will briefly highlight how they might be addressed in future work at the end of this chapter and also this thesis.

10.1.1 Leaning of Visual Classifiers versus Mappings in Semantics

The BURCHAK corpus (Yu et al., 2017b) contains some interesting conversations between humans (see an example in Table 10.2), where the learner is providing an irrelevant answer to the tutor's question. This might be because participants in this experiment (see Chapter 5) are required to describe visual attributes in a made-up language (e.g. "sako" for red and "burchak" for square), it is hard for the learner to group those invented words into different categories (colour and shape) without any clues, especially in the beginning of this task. In the experiment, the participant is required to translate existing (visual) knowledge to the made-up language.

However, unfortunately, such interesting conversations were filtered out from our work in this thesis. In contrast to that experiment, the proposed framework here has been assumed not learn the mappings between classifiers and semantics directly, but instead, learns visual classifiers themselves from scratch for new semantic items we might encounter within the dialogue. This means that, although the vision model, treating all visual attributes equally

Tutor:	hey, what is the colour and shape of this object?
Learner:	erm... a wakaki burchak?
Tutor:	no, wakaki is a shape, not colour. try again?
Learner:	oh sorry, sako?
Tutor:	yes, and shape?
Lerner:	burchak? sako burchak?
Tutor:	good job.

TABLE 10.2: An Irrelevant-Answering Conversation from the BURCHAK Corpus ("burchak" for square, "wakaki" for "triangle", "sako" for red)

as binary classifiers, does not identify the meaning of certain attribute words, the language/-dialogue module encodes grammars which know them and also their categories, for instance, in the grammar, "red" is assigned as a colour and "square" as a shape.

We were setting this work in such way because we noticed that visual knowledge and their mappings in a language are a pair of interactional factors [1] in a grounding problem, i.e. a good grounding performance must depend on the rich visual knowledge (pre-trained classifiers), like Kollar et al. (2013), Matuszek et al. (2012), Silberer and Lapata (2014), and vice versa, like our work here. As part of the future direction, we may attempt to improve this configuration, integrally learning both visual knowledge and their mappings, by extending the current framework with the theory of concept mapping by Wandersee (1990) and the probabilistic type theories by Cooper et al. (2014), Hough and Purver (2014a). The former theory, as a structural knowledge map, is designed to depict potential relationships between (cognitive) concepts. The latter is designed to construct TTR domain concept lattices with distributional probabilities over TTR candidates, with the aim of modelling incremental inference in dialogue.

10.2 Future Work

Following these discussions above, the work presented in the thesis is only the beginning rather than the end of the story. Apart from solving issues raised in above discussions, it can also be extended/improved in various ways:

- **Integrated Learning of Visual Concepts and Dialogue Strategies**:

  The current work is solving two problems, one for symbol grounding and another for dialogue management, individually. The multi-modal framework in the work is designed to learn the perceptual features based on a pre-trained optimised dialogue

[1] To our knowledge, the approach for jointly solving both factors (visual classifiers and their mappings in language) has not been implemented.

strategies, with a hand-crafted constrained state space in the Reinforcement Learning model. It means that the agent can easily learn groundings without external distractions, e.g. from language understanding and dialogue management. As part of future work, the situation can be improved for jointly learning both features, by deploying and extending the Deep Reinforcement Learning from Cuayáhuitl et al. (2016) and taking the raw-feature representation of the visual scene as input, and visual classifiers as the hidden layer.

- **Incremental Learning of New Visual Features and Descriptions**:

  In contrast with the given learning task, where visual-attribute words are limited on a very small scale (6 colours and 3 shapes), within human conversations in the real world, a simple visual scene (e.g. a person, object, attribute and event) can be described in different ways, especially while encountering unseen objects/features. For instance, sometimes the colour of "blue" can also be described as "cyan", or "it is like the sky". Hence, multi-modal distributional semantics by Bruni et al. (2014) can be introduced into this framework to address such challenges. His approach aims at generating word-embedding representations from both textual and visual resources, which can be used as features in a continuous space MDP to learn unseen attribute concepts.

- **Transferring the Multi-modal Framework to new Domains**:

  As mentioned at the beginning of this thesis, robots will be eventually brought out of the laboratory for various tasks. Different to a simple visual-attribute learning task, our framework can be transferred to other Human-Robot Collaboration domains and extended by incorporating with multi-modal approaches from Penkov et al. (2017) and Matuszek et al. (2014). Their approaches make use of different multi-modal information (deictic gestures by Matuszek et al. (2014) and eye-tracking by Penkov et al. (2017)) that enables robots to grounding symbols in an initially unknown environment and instantiate high-level plans to complete the task or commands autonomously.

- **Incremental Inference of Dialogue Act with the DS-TTR framework**:

  Although the DS-TTR model has been demonstrated to play an essential role on processing incremental dialogue, in this thesis, the current framework did not apply the model in a fully incremental process, i.e. the model employs a new inference mechanism that prevents the system from predicting the corresponding dialogue act until an utterance is completed parsed. In the future work, we can improve this mechanism to infer the user intent through an incremental word-by-word parsing process by also introducing the Probabilistic Type Theories by Cooper et al. (2014), Hough and Purver (2014a).

10.3 Final Word

Given the above, it seems that the full vision presented in this thesis has not been accomplished yet, but it does bring us a step closer to an answer about how the meaning of symbols (visual-attribute words) in a language can refer to the physical world incrementally, over time. Furthermore, the work in this thesis is not just a theoretical model, rather it has been applied to implement a visual and optimised interactive learning agent (VOILA) that is demonstrated and tested with human users in the real world.

Appendix A

Dynamic Syntax

The content about Dynamic Syntax in this Appendix is excerpted from Eshghi and Lemon (2014), Eshghi et al. (2011), Purver et al. (2011, 2014).

Dynamic Syntax (Cann et al., 2005b, Kempson et al., 2001) is a parsing-directed grammar formalism, which models the incremental, on-line processing of linguistic input. Unlike many other formalisms, DS models the incremental building up of *interpretations* without presupposing or indeed recognising an independent level of syntactic processing. Thus, the output for any given string of words, parsed incrementally, token by token, is a semantic tree (Eshghi et al., 2011) representing its predicate-argument structure, as in Figure 1.

John upset Mary.

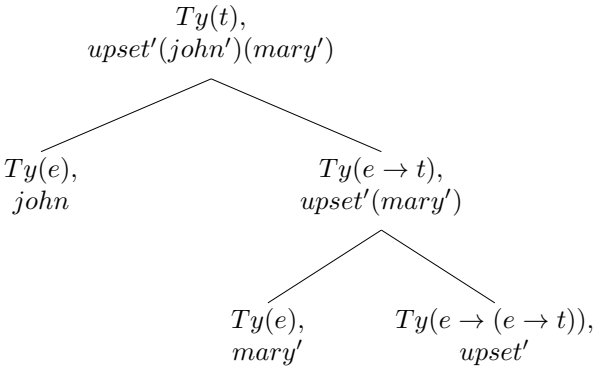


FIGURE A.1: A simple DS tree for *"John upset Mary"*

Grammaticality is defined as parsability, that is, the successful incremental construction of such tree-structure logical forms, using all the information given by the words so far in the sequence, in a left to right, time linear manner.

The logical formulae Eshghi et al. (2011) are lambda terms of the epsilon calculus (Meyer-Viol, 1995), and more recently Record Types of Type Theory with Records (TTR, see Cooper (2005), Eshghi et al. (2011) for how TTR has been integrated with DS).

A.1 Parsing

The central tree-growth process of the model is defined in terms of conditional *actions* (Eshghi et al., 2011) whereby such structures are built up. These take the form both of general structure-building principles (*computational actions*), independent of any particular natural language, and of specific actions induced by parsing particular lexical items (*lexical actions*). The core of the formal language is the modal tree logic LOFT (Blackburn and Meyer-Viol, 1994), which defines modal operators $\langle\downarrow\rangle$ and $\langle\uparrow\rangle$, which are interpreted as indicating daughter and mother relations, respectively, with two sub-cases $\langle\downarrow_0\rangle$, and $\langle\downarrow_1\rangle$ distinguishing daughters decorated with argument or functor formulae, and two additional operators $\langle L\rangle$ and $\langle L^{-1}\rangle$ to license paired linked trees (these are used for the interpretation of relative clauses and adjuncts among other things, see (Cann et al., 2005a)). These operators can be grouped together to specify different modalities (Eshghi et al., 2011), i.e. paths to other locations/nodes on the tree, relative to the current node. For example $\langle\uparrow_1\downarrow_0\rangle Ty(e)$ says that the argument sister of the current node is of type e. The actions defined using this language are conditional transition functions between semantic trees, monotonically extending the input tree and node decorations.

The node decorations are *labels* (Eshghi et al., 2011) of different kinds, carrying different types of information. In other words, each node is a set of labels. Labels can be true or false at any particular node; in most cases the truth of a label at a node is its presence on that node. One notable exception is that of modal labels, which usually check the presence of a label on another node on the tree, e.g. $\langle\uparrow_1\downarrow_0\rangle Ty(e)$ holds if $Ty(e)$ is present on the sister node of the current node.

The concept of requirement (also a kind of label (see Requirements in Eshghi et al. (2011))) is central to the parsing process, $?X$ representing the imposition of a goal to establish X, for any label X. Requirements may thus take the form $?Ty(t)$, $?Ty(e \rightarrow t)$, $?\langle\downarrow_1\rangle Ty(e \rightarrow t)$, $?\exists x Fo(x)$, $?\exists x Tn(x)$, etc. All requirements that are introduced have to be satisfied during the construction process.

For example, the first action in parsing a sentence is a general computational action (see more in Eshghi et al. (2011), Purver et al. (2011, 2014)), termed INTRODUCTION[1], which develops the standard initial Axiom tree, with only one root node (here, as in all such partial tree-structures, there is a pointer, $\diamond$, indicating the node under development):

[1]In more recent versions of DS, namely in the treatment of English auxiliaries in Cann (2010), the INTRODUCTION and PREDICTION actions are dispensed with for cross-linguistic generality, and also due to a unified treatment of passives, where the syntactic subject is fixed by the main verb itself to the logical object position.

$?Ty(t), Tn(0), \diamond$
(i.e. a basic requirement to construct a propositional formula), to
$?Ty(t)$, $Tn(0)$, $?\langle\downarrow_0\rangle Ty(e)$, $?\langle\downarrow_1\rangle Ty(e \rightarrow t)$, $\diamond$
thereby inducing the sub-goals of constructing a type e argument (0) node and a type $e \rightarrow t$ predicate (1) node, by which a predicate-argument formula can eventually be derived. In the lexicon, words are are associated with lexical actions (see the lexicon section in (Eshghi et al., 2011) for how the lexicon is structured) in a similar style, each a sequence of tree-update actions in an IF..THEN..ELSE format (Purver et al., 2011, 2014)), employing the explicitly procedural *atomic actions* (Eshghi et al., 2011), *make*, *go*, *put*, *merge* and others. A simple lexical action for a proper name *John* is as follows:

John	**IF**	$?Ty(e)$	If there's a requirement for a Type e formula
	THEN	$\texttt{put}(Ty(e))$	Label the node as Type e
		$\texttt{put}(Fo(John'))$	Label the node with semantic content/formula $John'$
		$\texttt{put}(\langle\downarrow\rangle \perp$	Bottom restriction: this is a leaf node.
	ELSE	ABORT	

FIGURE A.2: A simple lexical action for *"John"*

A subsequent general computational action (THINNING) then removes the now satisfied type requirement. A more complex lexical action for a transitive verb *dislike* takes the following form, first making a new predicate node of type $e \rightarrow (e \rightarrow t)$, and then an argument node with a requirement for type e (to be satisfied by parsing the object):

So an IF-THEN-ELSE action is composed of a sequence of labels, and two separate sequences of atomic actions (see Eshghi et al. (2011), Purver et al. (2011)), one comprising the THEN block, and the other the ELSE block. The execution of such an action proceeds by first *checking* (Eshghi et al., 2011) the labels in order; if all succeed (i.e. if all are true at the node where the pointer is), the THEN block is executed; otherwise the ELSE block of actions fire.

This format of lexical specification is general: all lexical items are defined as providing such actions, the concept of lexical content being essentially procedural. These obligatory lexical actions, together with optional computational actions (also in the same format), induce a sequence of partial trees in a monotonic growth relation as each word is consumed in turn.

We now turn to an algorithmic description of the parsing process.

dislike	**IF**	$?Ty(e \rightarrow t)$	If there's a requirement for a Type $e \rightarrow t$ formula
	THEN	$\texttt{make}(\langle\downarrow_1\rangle); \texttt{go}(\langle\downarrow_1\rangle)$	Make the functor (predicate) daughter (down 1) and go there
		$\texttt{put}(Ty(e \rightarrow (e \rightarrow t)))$	Label the node as Type $e \rightarrow (e \rightarrow t)$
		$\texttt{put}(Fo(\lambda x \lambda y.Dislike'(x)(y))$	Label the node with semantic content/formula $\lambda x \lambda y.Dislike'(x)(y)$
		$\texttt{put}(\langle\downarrow\rangle \perp)$ $\texttt{go}(\langle\uparrow_1\rangle)$	Bottom restriction: this is a leaf node. Go back up to the mother node
		$\texttt{make}(\langle\downarrow_0\rangle); \texttt{go}(\langle\downarrow_0\rangle); \texttt{put}(?Ty(e))$	Make the argument daughter, go there and label it as requiring a type e formula - which the object of the verb will provide at a later point in the parse.
	ELSE	ABORT	

FIGURE A.3: A simple lexical action for *"dislike"*

A.2 The parsing process

Given a sequence of words $(w_1, w_2, ..., w_n)$, the parser starts from the *axiom* tree T_0 (a requirement to construct a complete tree of type t), and applies the corresponding lexical actions $(a_1, a_2, \ldots, a_n)$, optionally interspersing general computational actions (which can apply whenever their preconditions are met). More precisely: we define the *parser state* (Eshghi et al., 2011) at step i as a set of partial trees S_i. Beginning with the singleton axiom state $S_0 = \{T_0\}$, for each word w_i:

1. Apply all lexical actions a_i corresponding to w_i to each partial tree in S_{i-1}. For each application that succeeds (i.e. the tree satisfies the action preconditions), add resulting (partial) tree to S_i.
2. *Adjust* the parse state: For each tree in S_i, apply all possible sequences of computational actions and add the result to S_i.

If at any stage the state S_i is empty, the parse has failed and the string is deemed ungrammatical. If the final state S_n contains a complete tree (all requirements satisfied), the

string is grammatical and its root node will provide the full sentence semantics; partial trees provide only partial semantic specifications.[2]

A.2.1 Graph representations

Sato (2011) shows how this procedure can be modelled as a *directed acyclic graph*, rooted at T_0, with individual partial trees as nodes, connected by edges representing single actions. While Sato uses this to model the search process, we exploit it (in a slightly modified form) to represent the linguistic *context* available during the parse – important in DS for ellipsis and pronominal construal (Details are given in (Cann et al., 2007, Gargett et al., 2009), but also see the next section A.2.2.).

We can also take a coarser-grained view via a graph which we term the *state* graph; here, nodes are states S_i and edges the sets of action sequences connecting them. This subsumes the tree graph, with state nodes containing possibly many tree-graph nodes; and here, nodes have multiple outgoing edges only when multiple word hypotheses are present. This corresponds directly to the input word graph (often called a word *lattice*) available from a speech recognizer, allowing close integration in a dialogue system – see below. We also see this as a suitable structure with which to begin to model phenomena such as hesitation and self-repair: as edges are linear action sequences, intended to correspond to the time-linear psycholinguistic processing steps involved, such phenomena may be analysed as building further edges from suitable departure points earlier in the graph.[3]

A.2.2 Parsing in Context

So far we have been considering parsing without any notion of context in place. This is, of course, essential if we are to account for context-dependent phenomena in dialogue or monologue, such as anaphora and different kinds of ellipsis, and their resolution. Here, we will not go into the detail of how exactly such constructions are analysed in DS (for which, see Cann et al. (2007)). Nevertheless, in order to later present how contextual parsing has been implemented we introduce the core mechanisms here.

There are generally three basic mechanisms in DS which enable the resolution of anaphora and ellipses from context:

[2]Note that only a subset of possible computational actions can apply to any given tree; together with a set of heuristics on possible application order, and the merging of identical trees produced by different sequences, this helps reduce complexity.

[3]There are similarities to chart parsing here: the tree graph edges spanning a state graph edge could be seen as corresponding to chart edges spanning a substring, with the tree nodes in the state S_i as the agenda. However, the lack of a notion of syntactic constituency means no direct equivalent for the active/passive edge distinction; a detailed comparison is still to be carried out.

1. *Substitution*:Copying/substitution of a formula from some other prior tree in context, into the tree under construction. This is used to deal with anaphora and strict VP-Ellipsis readings.

2. *Rerunning of Actions*:Rerunning a sequence of actions used before to build some prior tree in context, but this time with the current tree under construction as input. This is used to get sloppy readings for VP-Ellipsis, and to resolve some forms of Bare Argument Ellipsis (see Purver et al. (2006) for more theoretical detail)

3. *Direct Extension* Extending a tree in context directly: used to deal with Split Utterances and adjuncts but also Clarification Ellipsis and Sluicing.

Thus, we need to store both trees at every step of a parse, and the actions that were used to build them up, so that these may subsequently be used to recover the meaning of anaphoric and elliptical expressions. To achieve this, the notion of a parse state described above will be revised in favour of one in which parse states will have tuples in them rather than simply trees. We define a *parser tuple* (Eshghi et al., 2011) as a triple, $\langle T, W, A\rangle$, where T is a (possibly partial) semantic tree, W is the string of words so far parsed and A the sequence of actions (computational and lexical) used to construct T from W. The initial parse state S_0 contains only a single triple, in which T is the initial Axiom tree and W and A are both empty: $S_0 = \{\langle Axiom, \emptyset, \emptyset\rangle\}$. Accordingly we revise the parse process described above:

Given a sequence of words $(w_1, w_2, ..., w_n)$, the parser starts from the empty state $S_0 = \{\langle Axiom, \emptyset, \emptyset\rangle\}$, and applies the corresponding lexical actions $(a_1, a_2, \ldots, a_n)$, optionally interspersing general computational actions (which can apply whenever their preconditions are met). More precisely: we define the *parser state* (Eshghi et al., 2011) at step i as a set of parser tuples of the form $\langle T_i, W_i, A_i\rangle$. Beginning with $S_0 = \{\langle Axiom, \emptyset, \emptyset\rangle\}$, for each word w_i:

1. Apply all lexical actions a_i corresponding to w_i to each parser tuple $\langle T, W, A\rangle$ in S_{i-1}. If the application is successful (i.e. the input tree T satisfies the action preconditions), we will get an output tree T_o. Construct the new parser tuple $\langle T_o, W + w_i, A + a_i\rangle$ and add it to S_i.
2. *Adjust* (Eshghi et al., 2011)] the parse state: For each tree in S_i, apply all possible sequences of computational actions in the same manner and add the result to S_i.

If at any stage the state S_i is empty, the parse has failed and the string is deemed ungrammatical.

Context can now be defined in these terms. At any point in the parsing process, the context C for a particular partial tree T in the set S_i can be taken to consist of:

1. a set of triples $P' = \{..., \langle T_i, W_i, A_i \rangle, ...\}$ resulting from the previous utterance(s); and
2. the triple $\langle T, W, A \rangle$ itself.

Discourse-initially, the set P' will be empty, and the context will therefore be identical to the standard initial parser state, the singleton set P_0 containing only a single triple $\langle T_0, \emptyset, \emptyset \rangle$ (where T_0 is the basic Axiom $= \{?Ty(t), \diamond\}$, and the word and action sequences are empty). As words are consumed from the input string and the corresponding actions produce multiple possible partial trees, together with their corresponding word and action sequences, the parser state set will expand to contain multiple triples; note that the context C available to any tree will still be restricted to its current triple (as P' is empty at this point). Once parsing is complete, we use the final set P_1 to define the new starting state (and context) for the next sentence as $P_1 \cup P_0$ (i.e. P_1 with the addition of the triple containing the basic axiom). This is slightly different in the implementation for which see (Eshghi et al., 2011).

Note that here we have not covered how this context is in fact used to resolve ellipsis (for which see Purver et al. (2006)). Suffice to say that this is achieved by two special computational actions that use the context for tree update. These are SUBSTITUTION and REGENERATION. The former uses context to copy semantic content/formulae from, into the tree under construction and the latter develops the current tree using sequences of action stored in context. These are discussed in a bit more detail in Eshghi et al. (2011)

A.2.3 Generation

With the base formalism set out in a parsing perspective, we can define a generation system reflecting production that applies the very same parsing mechanism, as we shall see, leading to tight coordination between parsing and production. Our point of departure is Otsuka and Purver (2003), Purver and Otsuka (2003), which gives an initial method of context-independent tactical generation in which an output string is produced according to an input semantic tree, the goal tree. The generator incrementally produces a set of corresponding output strings and their associated partial trees (again, on a left-to-right, word-by-word basis) by following standard parsing routines and using the goal tree as a subsumption check. At each stage, partial strings and trees are tentatively extended using some word/action pair from the lexicon; only those candidates which subsume the goal tree are kept, and the process succeeds when a complete tree identical to the goal tree is produced (see Figure A.4). Generation and parsing thus use the same tree representations and tree-building actions throughout.

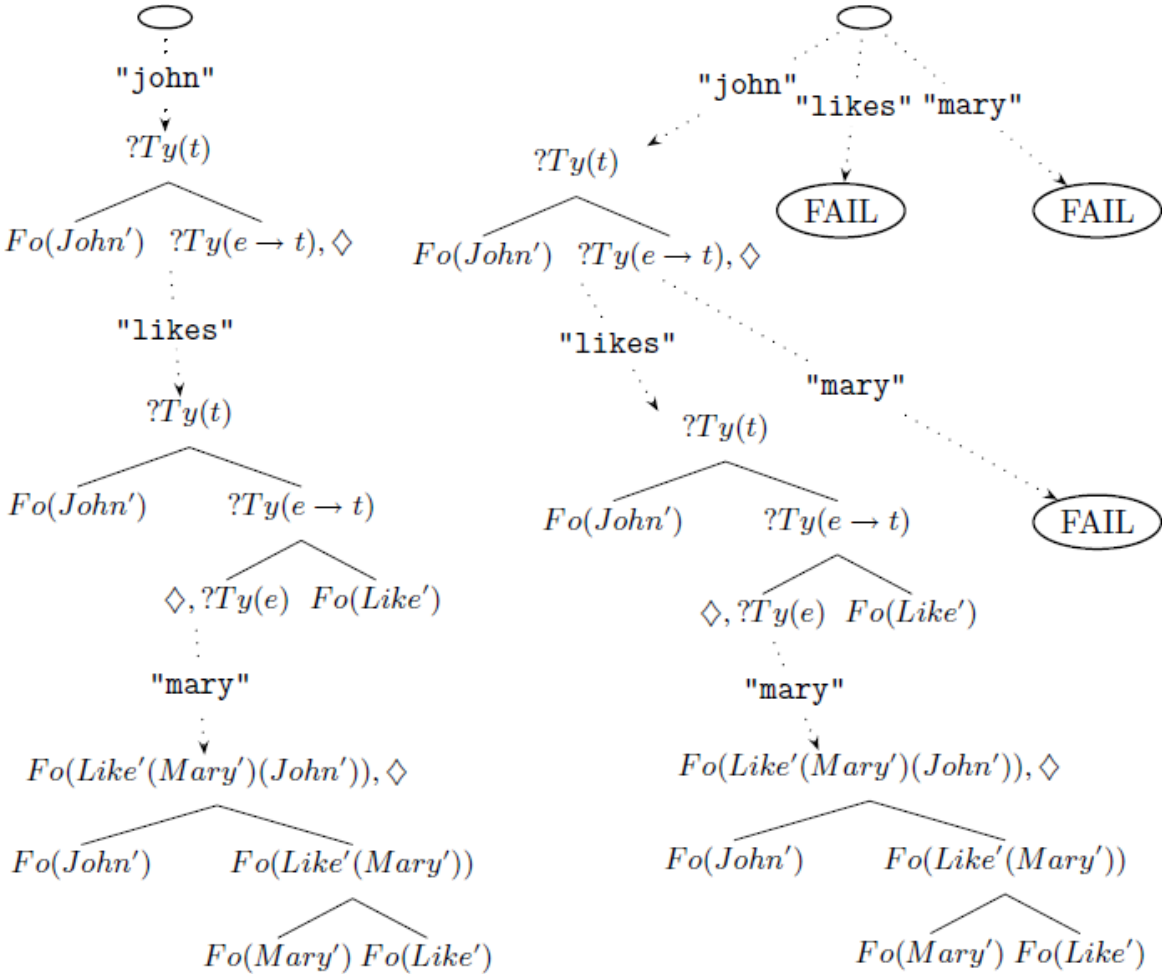


FIGURE A.4: Parsing (left) and Generating (right) of John likes Mary

We can proceed to the definition of a generator state. A generator state G is a pair (T_g, X) of a goal tree T_g and a set X of pairs (S, P), where S is a candidate partial string and P is the associated parser state (a set of $\langle T, W, A\rangle$ triples). Discourse-initially, the set X will contain only one pair, of an empty candidate string and the standard initial parser state, $(\emptyset, P_0)$. As generation progresses, multiple pairs are produced as candidate partial strings S are considered, each with their own associated parser state P. In generation, the context $\mathcal{C}$ for any partial tree T in a state P is defined exactly as for parsing: the set of triples $P' = \{..., \langle T_i, W_i, A_i\rangle, ...\}$; and the current triple $\langle T, W, A\rangle$. Once generation is complete, the state P_1 paired with the chosen string S_1 is taken to form the new context for the next sentence $P_1 \cup P_0$ (just as with parsing), hand-in-hand with the new initial generator state $X_1 = (\emptyset, P_1 \cup P_0)$. Note here the close relationship between the parsing and generation processes. They share the same basic component of their state (a parser state P, a set of tree/word-sequence/action-sequence triples the generator state merely adds to this (partial) candidate strings and a goal tree), and they share the same representation of context. In addition, as both processes are strictly incremental, there is no requirement that their initial

states be empty or contain only complete trees they can in theory start from any parser or generator state.

A.3 Integrating Type-Theory with Records (TTR)

More recent work in DS has started to explore the use of TTR to extend the formalism, replacing the atomic semantic type and FOL formula node labels with more complex *record types* (Eshghi et al., 2011), and thus providing a more structured semantic representation. This also allows tighter and more straightforward integration into incremental dialogue systems that work with concept frames, as the mapping between a concept frame and a TTR record type is straightforward (DyLan has been integrated into the Jindigo dialogue system (Schlangen and Skantze, 2009), for which see Eshghi et al. (2011)).

Purver et al. (2010) provide a sketch of one way to integrate TTR in DS and explain how it can be used to incorporate pragmatic information such as participant reference and illocutionary force. As shown in Figure A.5(b) above, we use a slightly different variant here: node record types are sequences of typed labels (e.g. $[x : e]$ for a label x of type e), with semantic content expressed by use of *manifest* types (e.g. $[x_{=john} : e]$ where $john$ is a singleton subtype of e).

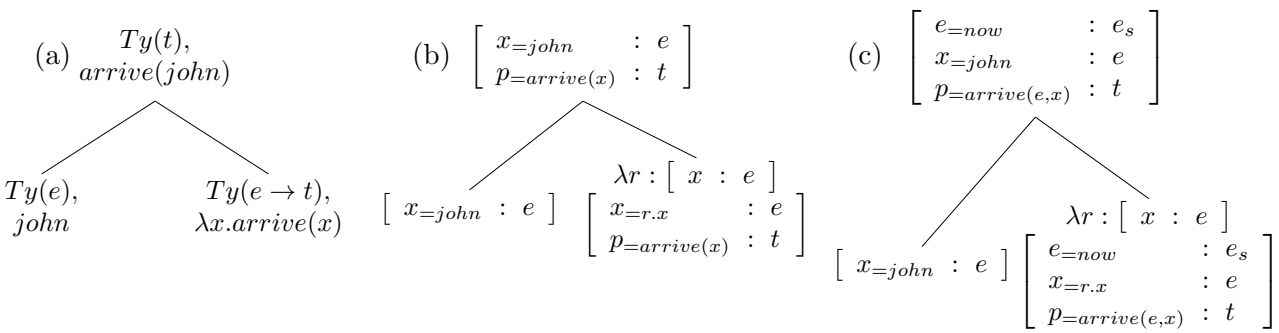


FIGURE A.5: A simple DS tree for *"john arrives"*: (a) original DS, (b) DS+TTR, (c) event-based

We further adopt an event-based semantics along Davidsonian lines (Davidson, 1980). As shown in Figure A.5(c), we include an event term (of type e_s) in the representation: this allows tense and aspect to be expressed (although Figure A.5(c) shows only a simplified version using the current time *now*). It also permits a straightforward analysis of optional adjuncts as extensions of an existing semantic representation; extensions which predicate over the event term already in the representation. Adding fields to a record type results in a more fully specified record type which is still a subtype of the original:

$$\begin{bmatrix} e_{=now} & : & e_s \\ x_{=john} & : & e \\ p_{=arrive(e,x)} & : & t \end{bmatrix} \quad \mapsto \quad \begin{bmatrix} e_{=now} & : & e_s \\ x_{=john} & : & e \\ p_{=arrive(e,x)} & : & t \\ p'_{=today(e)} & : & t \end{bmatrix}$$

"john arrives" $\mapsto$ *"john arrives today"*

FIGURE A.6: Optional adjuncts as leading to TTR subtypes

Appendix B

Instructions for the Human-Human Dialogue Data Collection

B.1 Consent Form

Thank you for agreeing to participate in todays experiment. At the end of the session you will be paid in £10 Amazon vouchers for your participation. In addition, at the end of each round of experimentation, the highest scoring pair will receive an additional £20.00 voucher each. In todays experiment, you will be communicating with another participant about the shape and colour of different objects. You will talk with each other using our custom-built chat tool, similar to WhatsApp or Facebook Messenger. Prior to the experiment you will be randomly assigned to one of two roles: Tutor or Learner. All data collected in this experiment will be anonymised and will be used for research purposes ONLY. You are free to stop participation at any point. Please do not hesitate to ask if you have any questions or problems.

[Consent Form]
Full Name:
Email:
Is English your first language? ☐ Yes ☐ No
☐ I have read and understood the procedures involved in the research and hereby consent to volunteer to participate in this study.
☐ I would like to participate further experiments for the same research.

Participants Signature: _____
Date: _____

B.2 Instructions for the Tutor

In this experiment, you, the Tutor, will be talking about the shape and colour of several objects to your partner, the Learner, using our custom built chat tool (similar to Whats App or Facebook Messenger). You will have some time to play around with the chat tool before you start the experiment.

The scenario:

Your task involves teaching your partner to identify the shape and colour of some simple objects (see Fig. 1), using new made-up words for the different colours and shapes. The shape and colour words will initially be new to both you and your partner. They are in fact not real words from the English language, but have been made up for this experiment only. Since you are the tutor, you will know these new words, but your partner will not. Your job is to teach them.

You, but not your partner, will be given a visual dictionary (see Fig. 1), where these new made-up colour and shape words are associated with their corresponding examples. You are not allowed to use any of the usual colour and shape words (e.g. red, circle, green) from the English language. You will be given a list of these banned words.

For example, when looking at a red square, you can say that it is tanager in colour, but you must not say that it is red.

The overall goal of the task is for your partner to correctly identify the colour and shape of as many objects as possible. In the beginning, your partner will not know any of the colour and shape words, but gradually over time, as you teach them, they should learn them and later be able to identify them in the objects presented later on in the task.

There are 9 objects in total, and it is up to you to move to the next object by clicking the next button. Note that you cannot go back to a previous object.

Scoring:

- For each colour or shape of each object correctly identified by the learner you will receive +1 point.
- Each time either you or your partner use a banned word from the list given to you, you will be penalised -0.5 point.
- If you cannot complete the task, i.e. go through all 9 objects in 30 minutes, you will be penalised -5 points

i.e. You will be assessed by the experimenter at the end of the experiment, and told how much you scored.

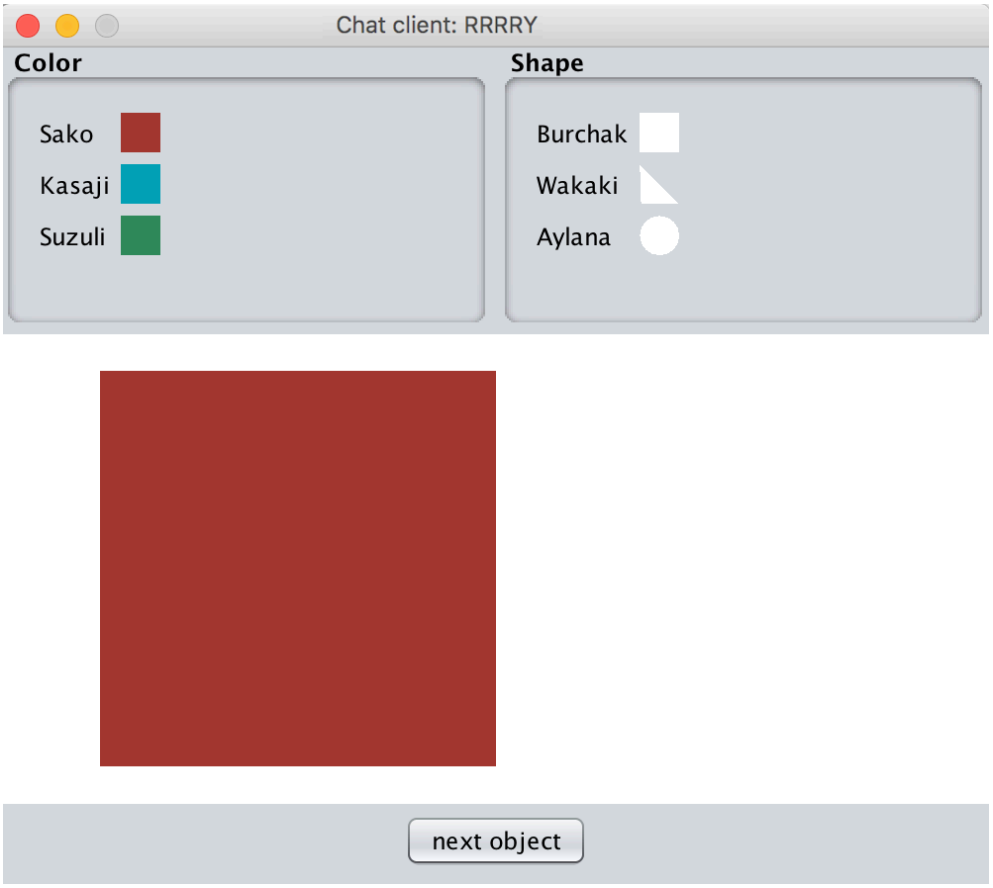


FIGURE B.1: Example Visual Dictionary together with The Current Object

List of Banned Words:

Red, green, yellow, pink, brown, blue etc. Triangle, triangular, circle, circular, ellipse, elliptical, rhombus, parallelogram, rectangle, rectangular, square, squared

B.3 Instructions for the Learner

In this experiment, you, the Learner, will be learning about the shape and colour of several objects from your partner, the Tutor, using our custom built chat tool (similar to Whats App or Facebook Messenger). You will have some time to play around with the chat tool before you start the experiment.

The scenario: Your task involves learning from your partner to identify the shape and colour of some very simple looking objects (see Fig. 1). The shape and colour words that you will learn about will initially be new to you. They are in fact not real words from the English language, but have been made up for this experiment only. Neither you nor your partner is allowed to use any of the usual colour and shape words (e.g. red, circle, green) from the English language. You will be given a list of these banned words.

For example you are allowed to ask Is it zifzaf? but not Is it red?

The overall goal of the task is for you to correctly identify the colour and shape of as many objects as possible. In the beginning, you will not know any of the new colour and shape words. But gradually, over time, your partner will teach you these words. As you learn more and more about what shapes and what colours these new words correspond to, you should be able to identify them in the objects presented later on in the task.

There are 9 objects in total. It is up to your partner, the tutor, to decide that you are done with the current object and that you can move to the next object. Note that your partner cannot and does not choose the objects, but can only move to the next one.

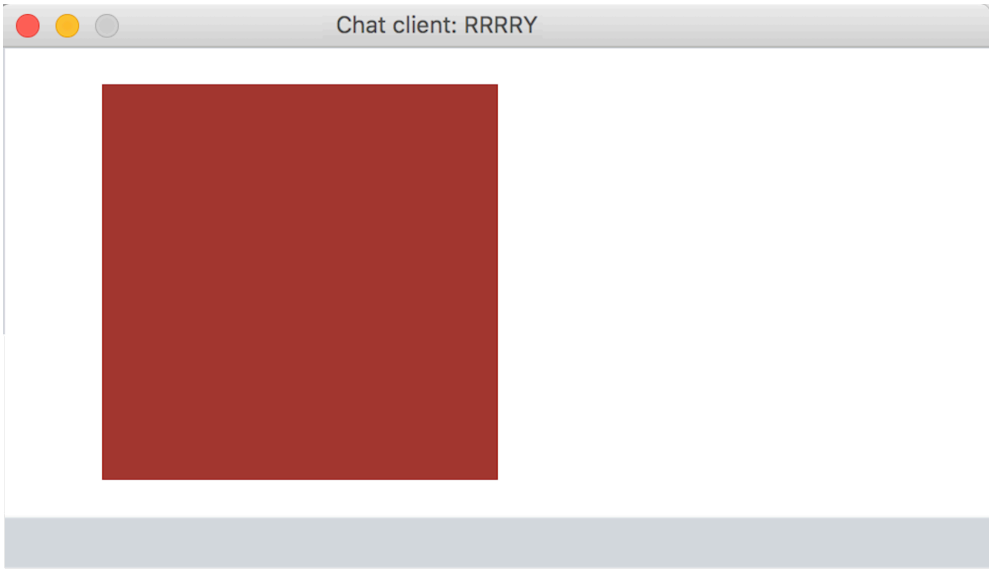


FIGURE B.2: Example Visual Dictionary together with The Current Object

Scoring:

- For each property of each object that you identify correctly without the tutor having told it to you on that specific object, you will receive +1 point.
- Each time either you or your partner use a banned word from the list given to you, you will be penalised -0.5 point.
- If you cannot complete the task, i.e. go through all 9 objects in 30 minutes, you will be penalised -5 points

i.e. You will be assessed by the experimenter at the end of the experiment, and told how much you scored.

List of Banned Words: Red, green, yellow, pink, brown, blue, etc. Triangle, triangular, circle, circular, ellipse, elliptical, rhombus, parallelogram, rectangle, rectangular, square, squared

Appendix C

Algorithm Pseudocode for Learning Dialogue Act Inference Grammar

In the Appendix, we present an algorithm pseudo-code of learning Dialogue Act inference grammars using the DS-TTR model.

Algorithm 1 Algorithm of Learning Dialogue Act Grammars

```
1: procedure LEARNING DA GRAMMARS THROUGH PRE-ANNOTATED DIALOGUES
2:     declare a map variable called da_inference_map that will contain a set of dialogue
   acts and corresponding grammar list;
3:     for each dialogue do
4:         initial the DyLan parser;    ▷ reset the parser at the beginning of each dialogue

5:         for each single utterance in dialogue do
6:             speaker = utterance.speaker;
7:             text = utterance;
8:             act = dialogueact;

9:             for each token in utterance text do      ▷ parse each utterance word-by-word
10:                PARSE(token);

11:            get original semantic tree from the Dylan parser;
12:            get the pointed node from the original semantic tree;
13:            get the TTR record type (ttr) from the pointed node;

14:            if the pointed node has tagged with act then
15:                skip to next utterance loop;
16:            else
17:                declare a new semantic tree called new_tree
18:                for each action-grammar in the existing list do
19:                    new_tree = execute action-grammar on semantic tree;

20:                if (new_tree is not tagged with act) or (new_tree is not defined) then
21:                    for each grammar template in grammar_templates based on act do
22:                        tree = execute the template on the semantic tree;
23:                        if tree is tagged with the act then
24:                            grammar template is selected;

25:                    if there is one template selected then
26:                        if there are inference grammars using same template then
27:                            for each learned inference grammar do
28:                                ttr_record = formula in action grammar;
29:                                if ttr_record is equal to ttr then
30:                                    skip to next utterance loop;

                                        ▷ there aren't inference grammar with the same formula
31:                        replace the actual attributes to META Variables in ttr;
32:                        add ttr into the selected template as a new grammar;
33:                        add new grammar into da_inference_maps;
34:                    else
35:                        throw an Unavailable Exception

36:    re-order da inference grammars based on the specificity of TTR record types.
```

Bibliography

Abdelsamea, M. M., Mohamed, M. H., and Bamatraf, M. (2015). An effective image feature classiffication using an improved SOM. *CoRR*, abs/1501.01723.

Afantenos, S., Asher, N., Benamara, F., Cadilhac, A., Degremont, C., Denis, P., Guhe, M., Keizer, S., Lascarides, A., Oliver Lemon, P. M., Paul, S., Rieser, V., and Vieu, L. (2012). Developing a corpus of strategic conversation in the settlers of catan. In *Proceedings of the 1st Workshop on Games and NLP*, Kanazawa, Japan.

Ai, H. and Weng, F. (2008). User simulation as testing for spoken dialog systems. In *Proceedings of the SIGDIAL 2008 Workshop, The 9th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 19-20 June 2008, Ohio State University, Columbus, Ohio, USA*, pages 164–171.

Allen, J., Ferguson, G., and Stent, A. (2001). An architecture for more realistic conversational systems. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces (IUI)*.

Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H., and Weinert, R. (1991). The HCRC map task data. *Language and Speech*, 34(4):351–366.

Asri, L. E., He, J., and Suleman, K. (2016). A sequence-to-sequence model for user simulation in spoken dialogue systems. *CoRR*, abs/1607.00070.

Aubrey, A. J., Marshall, A. D., Rosin, P. L., Vandeventer, J., Cunningham, D. W., and Wallraven, C. (2013). Cardiff conversation database (ccdb): A database of natural dyadic conversations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*, pages 277–282.

Baldi, P. (2002). A computational theory of surprise. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, pages 1–26.

Baumann, T., Kennington, C., Hough, J., and Schlangen, D. (2016). Recognising conversational speech: What an incremental asr should do for a dialogue system and how to

get there. In *International Workshop on Dialogue Systems Technology (IWSDS) 2016*. Universität Hamburg.

Belpaeme, T. (2001). Simulating the formation of color categories. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, pages 393–400.

Blackburn, P. and Meyer-Viol, W. (1994). Linguistics, logic and finite trees. *Logic Journal of the Interest Group of Pure and Applied Logics*, 2(1):3–29.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 421–436.

Brennan, S., Schuhmann, K., and Batres, K. (2013). Entrainment on the move and in the lab: The walking around corpus. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society, CogSci 2013, Berlin, Germany, July 31 - August 3, 2013*.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(1–47).

Buckley, M. and Wolska, M. (2008). A classification of dialogue actions in tutorial dialogue. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pages 73–80.

Bunt, H. (2006). Dimensions in dialogue act annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006.*, pages 919–924.

Buschmeier, H. and Kopp, S. (2013). Co-constructing grounded symbols–feedback and incremental adaptation in human-agent dialogue. *KI-Künstliche Intelligenz*, 27(2):137–143.

Buß, O., Baumann, T., and Schlangen, D. (2010). Collaborating on utterances with a spoken dialogue system using an ISU-based approach to incremental dialogue management. In *Proceedings of the SIGDIAL 2010 Conference*, pages 233–236, Tokyo, Japan. Association for Computational Linguistics.

Buß, O. and Schlangen, D. (2011). Dium–an incremental dialogue manager that can produce self-corrections. *Proceedings of SemDial 2011 (Los Angelogue)*.

Cann, R. (2010). Towards an account of the english auxiliary system. In Gregoromichelaki, E., Kempson, R., and Howes, C., editors, *The Dynamics of Lexical Interfaces*. CSLI. to appear.

Cann, R., Kaplan, T., and Kempson, R. (2005a). Data at the grammar-pragmatics interface: the case of resumptive pronouns in English. *Lingua*, 115(11):1475–1665. Special Issue: On the Nature of Linguistic Data.

Cann, R., Kempson, R., and Marten, L. (2005b). *The Dynamics of Language*. Elsevier, Oxford.

Cann, R., Kempson, R., and Purver, M. (2007). Context and well-formedness: the dynamics of ellipsis. *Research on Language and Computation*, 5(3):333–358.

Chai, J. Y., Fang, R., Liu, C., and She, L. (2016). Collaborative language grounding toward situated human-robot dialogue. *AI Magazine*, 37(4):32–45.

Chandramohan, S., Geist, M., Lefevre, F., and Pietquin, O. (2012). Behavior specific user simulation in spoken dialogue systems. In *Speech Communication; 10. ITG Symposium; Proceedings of*, pages 1–4. VDE.

Chung, G. (2004). Developing a flexible spoken dialog system using simulation. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*, pages 63–70.

Clark, H. H. (1996). *Using Language*. Cambridge University Press.

Clark, H. H. and Brennan, S. A. (1991). *Grounding in communication*, pages 127–149. Washington: APA Books.

Clark, H. H. and Fox Tree, J. E. (2002). Using *uh* and *um* in spontaneous speaking. *Cognition*, 84(1):73–111.

Colman, M. and Healey, P. G. T. (2011). The distribution of repair in dialogue. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, pages 1563–1568, Boston, MA.

Cooper, R. (2005). Records and record types in semantic theory. *Journal of Logic and Computation*, 15(2):99–112.

Cooper, R. (2012). Type theory and semantics in flux. In Kempson, R., Asher, N., and Fernando, T., editors, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics, pages 271–323. North Holland.

Cooper, R., Dobnik, S., Lappin, S., and Larsson, S. (2014). A probabilistic rich type theory for semantic interpretation. In *Proceedings of the EACL Workshop on Type Theory and Natural Language Semantics (TTNLS)*, Gothenburg, Sweden. Association for Computational Linguistics.

Cooper, R. and Ginzburg, J. (2015). Type theory with records for natural language semantics. In Lappin, S. and Fox, C., editors, *Handbook of Contemporary Semantic Theory (second edition)*, pages 375–407. Wiley-Blackwell.

Copestake, A. (2007). Semantic composition with (robust) minimal recursion semantics. In *Proceedings of the ACL-07 workshop on Deep Linguistic Processing*, pages 73–80.

Crocker, M., Pickering, M., and Clifton, C., editors (2000). *Architectures and Mechanisms in Sentence Comprehension*. Cambridge University Press.

Cuayáhuitl, H., Renals, S., Lemon, O., and Shimodaira, H. (2005). Human-computer dialogue simulation using hidden markov models. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 290–295. IEEE.

Cuayáhuitl, H., Yu, S., Williamson, A., and Carse, J. (2016). Deep reinforcement learning for multi-domain dialogue systems. *CoRR*, abs/1611.08675.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M. F., Parikh, D., and Batra, D. (2016). Visual dialog. *CoRR*, abs/1611.08669.

Das, A., Kottur, S., Moura, J. M. F., Lee, S., and Batra, D. (2017). Learning cooperative visual dialog agents with deep reinforcement learning. *CoRR*, abs/1703.06585.

Davidson, D. (1980). *Essays on Actions and Events*. Clarendon Press, Oxford, UK.

de Vries, H., Strub, F., Chandar, S., Pietquin, O., Larochelle, H., and Courville, A. C. (2016). Guesswhat?! visual object discovery through multi-modal dialogue. *CoRR*, abs/1611.08481.

Dobnik, S., Cooper, R., and Larsson, S. (2012a). Modelling language, action, and perception in type theory with records. In *Proceedings of the 7th International Workshop on Constraint Solving and Language Processing (CSLP?Äô12)*, pages 51–63.

Dobnik, S., Cooper, R., and Larsson, S. (2012b). Modelling language, action, and perception in type theory with records. In *Constraint Solving and Language Processing - 7th International Workshop, CSLP 2012, Orléans, France, September 13-14, 2012, Revised Selected Papers*, pages 70–91.

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Darrell, T., and Saenko, K. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2625–2634.

Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 80–87. IEEE.

Eshghi, A. (2015). DS-TTR: An incremental, semantic, contextual parser for dialogue. In *Proceedings of Semdial 2015 (goDial), the 19th workshop on the semantics and pragmatics of dialogue*.

Eshghi, A. and Healey, P. G. T. (2015). Collective contexts in conversation: Grounding by proxy. *Cognitive Science*, pages 1–26.

Eshghi, A., Hough, J., and Purver, M. (2013). Incremental grammar induction from child-directed dialogue utterances. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 94–103, Sofia, Bulgaria. Association for Computational Linguistics.

Eshghi, A., Hough, J., Purver, M., Kempson, R., and Gregoromichelaki, E. (2012). Conversational interactions: Capturing dialogue dynamics. In Larsson, S. and Borin, L., editors, *From Quantification to Conversation: Festschrift for Robin Cooper on the occasion of his 65th birthday*, volume 19 of *Tributes*, pages 325–349. College Publications, London.

Eshghi, A., Howes, C., Gregoromichelaki, E., Hough, J., and Purver, M. (2015). Feedback in conversation as incremental semantic update. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS 2015)*, London, UK. Association for Computational Linguisitics.

Eshghi, A. and Lemon, O. (2014). How domain-general can we be? learning incremental dialogue systems without dialogue acts. In *Proceedings of SemDial*.

Eshghi, A., Purver, M., and Hough, J. (2011). Dylan: Parser for dynamic syntax. Technical report, Queen Mary University of London.

Eshghi, A., Shalyminov, I., and Lemon, O. (2017). Bootstrapping incremental dialogue systems from minimal data: linguistic knowledge or machine learning? In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Eshky, A., Allison, B., and Steedman, M. (2012). Generative goal-driven user simulation for dialog management. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 71–81.

Farhadi, A., Endres, I., and Hoiem, D. (2010). Attribute-centric recognition for cross-category generalization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2352–2359. IEEE.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR.*

Fernández, R. (2006). *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. PhD thesis, King's College London, University of London.

Ferreira, V. (1996). Is it better to give than to donate? Syntactic flexibility in language production. *Journal of Memory and Language*, 35:724–755.

Frank, E., Hall, M. A., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., and Trigg, L. (2010). Weka-a machine learning workbench for data mining. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pages 1269–1277.

Fraser, N. M. and Gilbert, G. N. (1991). Simulating speech systems. *Computer Speech & Language*, 5(1):81–99.

Fu, Y., Yang, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2014). Transductive multi-label zero-shot learning. In *BMVC*.

Furao, S. and Hasegawa, O. (2006). An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106.

Gargett, A., Gregoromichelaki, E., Kempson, R., Purver, M., and Sato, Y. (2009). Grammar resources for modelling dialogue dynamically. *Cognitive Neurodynamics*, 3(4):347–363.

Gasic, M., Breslin, C., Henderson, M., Kim, D., Szummer, M., Thomson, B., Tsiakoulis, P., and Young, S. J. (2013). Pomdp-based dialogue manager adaptation to extended domains. In *Proceedings of the SIGDIAL 2013 Conference, The 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 22-24 August 2013, SUPELEC, Metz, France*, pages 214–222.

Gasic, M. and Young, S. J. (2014). Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 22(1):28–40.

Georgila, K., Henderson, J., and Lemon, O. (2005). Learning user simulations for information state update dialogue systems. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*, pages 893–896.

Georgila, K., Henderson, J., and Lemon, O. (2006). User simulation for spoken dialogue systems: learning and evaluation. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006.*

Ghigi, F., Eskénazi, M., Torres, M. I., and Lee, S. (2014). Incremental dialog processing in a task-oriented dialog. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 308–312.

Ginzburg, J. (2012). *The Interactive Stance: Meaning for Conversation*. Oxford University Press.

Godfrey, J. J., Holliman, E., and McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of IEEE ICASSP-92*, pages 517–520, San Francisco, CA.

Greco, A. and Carrea, E. (2012). Grounding compositional symbols: no composition without discrimination. *Cognitive Processing*, 13(2):139–150.

Harnad, S. (1999). The symbol grounding problem. *CoRR*, cs.AI/9906002.

Healey, P. G. T., Purver, M., King, J., Ginzburg, J., and Mills, G. (2003). Experimenting with clarification in dialogue. In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*, Boston, Massachusetts.

Heinroth, T. and Minker, W. (2012). *Introducing spoken dialogue systems into Intelligent Environments*. Springer Science & Business Media.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Hough, J. (2011). Incremental semantics driven natural language generation with self-repairing capability. In *Recent Advances in Natural Language Processing (RANLP)*, pages 79–84, Hissar, Bulgaria.

Hough, J. (2014). *Modelling incremental self-repair processing in dialogue*. PhD thesis, Queen Mary University of London, UK.

Hough, J. (2015). *Modelling Incremental Self-Repair Processing in Dialogue*. PhD thesis, Queen Mary University of London.

Hough, J. and Purver, M. (2012). Processing self-repairs in an incremental type-theoretic dialogue system. In *Proceedings of the 16th SemDial Workshop on the Semantics and Pragmatics of Dialogue (SeineDial)*, pages 136–144, Paris, France.

Hough, J. and Purver, M. (2014a). Probabilistic type theory for incremental dialogue processing. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 80–88, Gothenburg, Sweden. Association for Computational Linguistics.

Hough, J. and Purver, M. (2014b). Strongly incremental repair detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics. to appear.

Hough, J. and Schlangen, D. (2017). Joint, Incremental Disfluency Detection and Utterance Segmentation from Speech. In *Proceedings of the Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 326–336.

Howes, C. and Eshghi, A. (2017). Feedback relevance spaces: the organisation of increments in conversation. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

Howes, C., Healey, P. G. T., Purver, M., and Eshghi, A. (2012). Finishing each other's ... responding to incomplete contributions in dialogue. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society (CogSci 2012)*, pages 479–484, Sapporo, Japan.

Howes, C., Purver, M., Healey, P. G. T., Mills, G. J., and Gregoromichelaki, E. (2011). On incrementality in dialogue: Evidence from compound contributions. *Dialogue and Discourse*, 2(1):279–311.

Janícek, M. (2011). Abductive reasoning for continual dialogue understanding. In *New Directions in Logic, Language and Computation - ESSLLI 2010 and ESSLLI 2011 Student Sessions. Selected Papers*, pages 16–31.

Jung, S., Lee, C., Kim, K., Jeong, M., and Lee, G. G. (2009). Data-driven user simulation for automated evaluation of spoken dialog systems. *Computer Speech & Language*, 23(4):479–509.

Jung, S., Lee, C., Kim, K., Lee, D., and Lee, G. G. (2011). Hybrid user intention modeling to diversify dialog simulations. *Computer Speech & Language*, 25(2):307–326.

Kalatzis, D., Eshghi, A., and Lemon, O. (2016a). Bootstrapping incremental dialogue systems: using linguistic knowledge to learn from minimal data. In *Proceedings of the NIPS 2016 workshop on Learning Methods for Dialogue*, Barcelona.

Kalatzis, D., Eshghi, A., and Lemon, O. (2016b). Bootstrapping incremental dialogue systems: using linguistic knowledge to learn from minimal data. *CoRR*, abs/1612.00347.

Karpathy, A. and Fei-Fei, L. (2014). Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306.

Karpathy, A. and Li, F. (2015). Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137.

Keizer, S., Rossignol, S., Chandramohan, S., and Pietquin, O. (2012). User Simulation in the Development of Statistical Spoken Dialogue Systems. In Pietquin, O. L. O., editor, *Data-Driven Methods for Adaptive Spoken Dialogue Systems: Computational Learning for Conversational Interfaces*, chapter 4, pages 39–73. Springer.

Kempson, R., Cann, R., Eshghi, A., Gregoromichelaki, E., and Purver, M. (2015). Ellipsis. In Lappin, S. and Fox, C., editors, *The Handbook of Contemporary Semantics*. Wiley-Blackwell.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Kennington, C. (2016). *Incrementally resolving references in order to identify visually present objects in a situated dialogue setting*. PhD thesis.

Kennington, C., Dia, L., and Schlangen, D. (2015). A discriminative model for perceptually-grounded incremental reference resolution. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 195–205.

Kennington, C., Kousidis, S., and Schlangen, D. (2014). Inprotks: A toolkit for incremental situated processing. In *Proceedings of the SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 84–88.

Kennington, C. and Schlangen, D. (2014). Situated incremental natural language understanding using markov logic networks. *Computer Speech & Language*, 28(1):240–255.

Kennington, C. and Schlangen, D. (2015). Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 292–301.

Kim, S., D'Haro, L. F., Banchs, R. E., Williams, J., and Henderson, M. (2016). The Fourth Dialog State Tracking Challenge. In *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*.

Kimura, D., Nishimura, R., Oguro, A., and Hasegawa, O. (2013). Ultra-fast multimodal and online transfer learning on humanoid robots. In *ACM/IEEE International Conference on Human-Robot Interaction, HRI 2013, Tokyo, Japan, March 3-6, 2013*, pages 165–166.

Kiros, R., Salakhutdinov, R., and Zemel, R. (2014). Multimodal neural language models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 595–603.

Kollar, T., Krishnamurthy, J., and Strimel, G. (2013). Toward interactive grounded language acqusition. In *Robotics: Science and Systems*.

Kong, X., Ng, M. K., and Zhou, Z. (2013). Transductive multilabel learning via label set propagation. *IEEE Trans. Knowl. Data Eng.*, 25(3):704–719.

Kristan, M. and Leonardis, A. (2014). Online discriminative kernel density estimator with gaussian kernels. *IEEE Trans. Cybernetics*, 44(3):355–365.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465.

Lansdale, M. W. and Ormerod, T. C. (1994). *Understanding interfaces - a handbook of human-computer dialogue*. Computers and people series. Academic Press.

Larsson, S. (2002). *Issue-based Dialogue Management*. PhD thesis, Göteborg University. Also published as Gothenburg Monographs in Linguistics 21.

Larsson, S. (2013). Formal semantics for perceptual classification. *Journal of logic and computation*.

Larsson, S. (2015). Formal semantics for perceptual classification. *J. Log. Comput.*, 25(2):335–369.

Leech, G. (1992). 100 million words of english: the british national corpus (bnc). *IEEE Trans. Affective Computing*, 28(1):1–13.

Levin, E., Pieraccini, R., and Eckert, W. (2000). A stochastic model of human-machine interaction for learning dialogue strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1).

Li, J. and Dey, S. (2013). Wizard of oz in human computer interaction.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Lison, P. (2014). *Structured Probabilistic Modelling for Dialogue Management*. PhD thesis, University of Oslo, Norway.

Louwerse, M. M. and Crossley, S. A. (2006). Dialog act classification using n-gram algorithms. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*, pages 758–763.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 285–294.

Matuszek, C., Bo, L., Zettlemoyer, L., and Fox, D. (2014). Learning from unscripted deictic gesture and language for human-robot interactions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2556–2563.

Matuszek, C., FitzGerald, N., Zettlemoyer, L., Bo, L., and Fox, D. (2012). A joint model of language and perception for grounded attribute learning. In *Proc. of the 2012 International Conference on Machine Learning*, Edinburgh, Scotland.

Meyer-Viol, W. (1995). *Instantial Logic*. PhD thesis, University of Utrecht.

Mills, G. J. and Healey, P. G. T. (2017). The Dialogue Experimentation toolkit. *(Submitted)*.

Mitchell, C. M., Ha, E., Boyer, K. E., and Lester, J. C. (2012). Recognizing effective and student-adaptive tutor moves in task-oriented tutorial dialogue. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, Marco Island, Florida. May 23-25, 2012.*

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 689–696.

Otsuka, M. and Purver, M. (2003). Incremental generation by incremental parsing. In *Proceedings of the 6th CLUK Colloquium*, pages 93–100, Edinburgh. CLUK.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318.

Penkov, S., Bordallo, A., and Ramamoorthy, S. (2017). Physical symbol grounding and instance learning through demonstration and eye tracking. In *Robotics and Automation, 2017 IEEE International Conference on*, Singapore.

Purver, M., Cann, R., and Kempson, R. (2006). Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2-3):289–326.

Purver, M., Eshghi, A., and Hough, J. (2011). Incremental semantic construction in a dialogue system. In Bos, J. and Pulman, S., editors, *Proceedings of the 9th International Conference on Computational Semantics (IWCS)*, pages 365–369, Oxford, UK.

Purver, M., Fernández, R., Frampton, M., and Peters, S. (2009a). Cascaded lexicalised classifiers for second-person reference resolution. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009 Conference)*, pages 306–309, London, UK. Association for Computational Linguistics.

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., and Cann, R. (2010). Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar. In Łupkowski, P. and Purver, M., editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, pages 43–50, Poznań. Polish Society for Cognitive Science.

Purver, M., Hough, J., and Gregoromichelaki, E. (2014). Dialogue and compound contributions. In Bangalore, S. and Stent, A., editors, *Natural Language Generation in Interactive Systems*, pages 63–92. Cambridge University Press.

Purver, M., Howes, C., Gregoromichelaki, E., and Healey, P. G. T. (2009b). Split utterances in dialogue: A corpus study. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL 2009 Conference)*, pages 262–271, London, UK. Association for Computational Linguistics.

Purver, M. and Otsuka, M. (2003). Incremental generation by incremental parsing: Tactical generation in Dynamic Syntax. In *Proceedings of the 9th European Workshop in Natural Language Generation (ENLG)*, pages 79–86.

Riek, L. D. (2012). Wizard of oz studies in hri: A systematic review and new reporting guidelines. *J. Hum.-Robot Interact.*, 1(1):119–136.

Rieser, V. (2008). *Bootstrapping reinforcement learning-based dialogue strategies from wizard-of-oz data*. PhD thesis, Saarland University.

Rieser, V. and Lemon, O. (2006). Cluster-based user simulations for learning dialogue strategies. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*.

Rougier, N. P. (2009). Implicit and explicit representations. *Neural Networks*, 22(2):155–160.

Roy, D. (2002). A trainable visually-grounded spoken language generation system. In *Proceedings of the International Conference of Spoken Language Processing*.

Sato, Y. (2011). Local ambiguity, search strategies and parsing in Dynamic Syntax. In Gregoromichelaki, E., Kempson, R., and Howes, C., editors, *The Dynamics of Lexical Interfaces*. CSLI Publications.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M., and Strope, B. (2010). "Your word is my command": Google search by voice: A case study. In *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*, chapter 4, pages 61–90. Springer, New York.

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., and Young, S. J. (2007a). Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 149–152.

Schatzmann, J., Thomson, B., and Young, S. J. (2007b). Error simulation for training statistical dialogue systems. In *IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2007, Kyoto, Japan, December 9-13, 2007*, pages 526–531.

Schatzmann, J., Thomson, B., and Young, S. J. (2007c). Statistical user simulation with a hidden agenda. In *Proceedings of the SIGDIAL 2007 Workshop, The 9th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 1-2 September 2007, the University of Antwerp, Belgium*.

Schlangen, D. (2016). Grounding, justification, adaptation: Towards machines that mean what they say. In *Proceedings of the 20th Workshop on the Semantics and Pragmatics of Dialogue (JerSem)*.

Schlangen, D. and Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 710–718, Athens, Greece. Association for Computational Linguistics.

Schlangen, D. and Skantze, G. (2011). A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111.

Selfridge, E., Arizmendi, I., Heeman, P. A., and Williams, J. D. (2012). Integrating incremental speech recognition and pomdp-based dialogue systems. In *Proceedings of the SIGDIAL 2012 Conference, The 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 5-6 July 2012, Seoul National University, Seoul, South Korea*, pages 275–279.

Serban, I. V., Lowe, R., Charlin, L., and Pineau, J. (2015). A survey of available corpora for building data-driven dialogue systems. *CoRR*, abs/1512.05742.

Shepard, R. N. and Cooper, L. A. (1986). Mental images and their transformations. *The American Journal of Psychology*, 96.

Silberer, C., Ferrari, V., and Lapata, M. (2013). Models of semantic representation with visual attributes. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 572–582, Sofia, Bulgaria. Association for Computational Linguistics.

Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

Sirinart Tangruamsub, A. K. and Hasegawa, O. (2011). Fast online incremental attribute-based object classification using stochastic gradient descent and self-organizing incremental neural network. In *Image Processing and Computer Vision (IPCV)*, USA.

Skantze, G. (2007). *Error Handling in Spoken Dialogue Systems : Managing Uncertainty, Grounding and Miscommunication*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden.

Skantze, G. and Hjalmarsson, A. (2010). Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan. Association for Computational Linguistics.

Skocaj, D., Kristan, M., Vrecko, A., Mahnic, M., Janícek, M., Kruijff, G. M., Hanheide, M., Hawes, N., Keller, T., Zillich, M., and Zhou, K. (2011). A system for interactive learning in dialogue with a tutor. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2011, San Francisco, CA, USA, September 25-30, 2011*, pages 3387–3394.

Skocaj, D., Vrecko, A., Mahnic, M., Janícek, M., Kruijff, G. M., Hanheide, M., Hawes, N., Wyatt, J. L., Keller, T., Zhou, K., Zillich, M., and Kristan, M. (2016). An integrated system for interactive continuous learning of categorical knowledge. *J. Exp. Theor. Artif. Intell.*, 28(5):823–848.

Skočaj, D., Kristan, M., and Leonardis, A. (2009). Formalization of different learning strategies in a continuous learning framework. In *Proceedings of the Ninth International Conference on Epigenetic Robotics; Modeling Cognitive Development in Robotic Systems*, pages 153–160. Lund University Cognitive Studies.

Socher, R., Karpathy, A., Le, Q. V., Manning, C. D., and Ng, A. Y. (2014). Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.

Steedman, M. (1991). Type-raising and directionality in combinatory grammar. In *29th Annual Meeting of the Association for Computational Linguistics, 18-21 June 1991, University of California, Berkeley, California, USA, Proceedings.*, pages 71–78.

Steels, L. (2008). *The symbol grounding problem has been solved, so what?s next?*

Steels, L., Belpaeme, T., et al. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and brain sciences*, 28(4):469–488.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Ess-Dykema, C. V., Martin, R., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Strub, F., de Vries, H., Mary, J., Piot, B., Courville, A. C., and Pietquin, O. (2017). End-to-end optimization of goal-driven and visually grounded dialogue systems. *CoRR*, abs/1703.05423.

Sun, Y., Bo, L., and Fox, D. (2013). Attribute based object identification. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2096–2103. IEEE.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: an Introduction*. MIT Press.

Tellex, S., Thaker, P., Joseph, J. M., and Roy, N. (2014). Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, 94(2):151–167.

Tellex, S., Thakerll, P., Deitsl, R., Simeonovl, D., Kollar, T., and Royl, N. (2013). Toward information theoretic human-robot dialog. *Robotics: Science and Systems*, page 409.

Thomason, J., Sinapov, J., Sevtlik, M., Stone, P., and Mooney, R. J. (2016a). Learning multi-modal grounded linguistic semantics by playing "i spy". In *To Appear: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-16, New York City, USA, July 9-15, 2016*.

Thomason, J., Sinapov, J., Sevtlik, M., Stone, P., and Mooney, R. J. (2016b). Learning multi-modal grounded linguistic semantics by playing "i spy". In *To Appear: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-16, New York City, USA, July 9-15, 2016*.

Thomason, J., Zhang, S., Mooney, R. J., and Stone, P. (2015). Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1923–1929.

Thompson, H. S., Anderson, A., Bard, E. G., Doherty-Sneddon, G., Newlands, A., and Sotillo, C. (1993). The hcrc map task corpus: Natural dialogue for speech recognition. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 25–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Traum, D. and Larsson, S. (2003). The information state approach to dialogue management. In Smith and Kuppevelt, editors, *Current and New Directions in Discourse & Dialogue*, pages 325–353. Kluwer Academic Publishers.

Venugopalan, S., Hendricks, L. A., Mooney, R. J., and Saenko, K. (2016). Improving lstm-based video description with linguistic knowledge mined from text. *CoRR*, abs/1604.01729.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R. J., and Saenko, K. (2014). Translating videos to natural language using deep recurrent neural networks. *CoRR*, abs/1412.4729.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164.

Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. (1997). Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 271–280, Madrid, Spain. Association for Computational Linguistics.

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., Wolf, P., and Woelfel, J. (2004). Sphinx-4: A flexible open source framework for speech recognition. Technical report, Mountain View, CA, USA.

Wandersee, J. H. (1990). Concept mapping and the cartography of cognition. *Journal of research in science teaching*, 27(10):923–936.

Webb, N. (2010). *Cue-based dialogue act classification*. PhD thesis, University of Sheffield, UK.

Weston, J., Bordes, A., Chopra, S., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Whitney, D., Rosen, E., MacGlashan, J., and L.S. Wong, Lawson and, T. S. (2017). Reducing errors in object-fetching interactions through social feedback. In *Proceedings of the IEEE International Conference on Robotics and Automation ICRA 2017, May 29 – June 3, 2017, Marina Bay Sands, Singapore*.

Williams, J. and Young, S. (2007). Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116–2129.

Wittgenstein, L. (1953). *Philosophical investigations*. Wiley-Blackwell.

Yu, Y., Eshghi, A., and Lemon, O. (2015a). Comparing attribute classifiers for interactive language grounding. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 60–69, Lisbon, Portugal. Association for Computational Linguistics.

Yu, Y., Eshghi, A., and Lemon, O. (2016a). Incremental generation of visually grounded language in situated dialogue (demonstration system). In *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK*, pages 109–110.

Yu, Y., Eshghi, A., and Lemon, O. (2016b). Interactively learning visually grounded word meanings from a human tutor. In *Proceedings of the 5th Workshop on Vision and Language, hosted by the 54th Annual Meeting of the Association for Computational Linguistics, VL@ACL 2016, August 12, Berlin, Germany*.

Yu, Y., Eshghi, A., and Lemon, O. (2016c). Training an adaptive dialogue policy for interactive learning of visually grounded word meanings. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 339–349.

Yu, Y., Eshghi, A., and Lemon, O. (2017a). Learning how to learn: An adaptive dialogue agent for incrementally learning visually grounded word meanings. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 10–19, Vancouver, Canada. Association for Computational Linguistics.

Yu, Y., Eshghi, A., Mills, G., and Lemon, O. (2017b). The burchak corpus: a challenge data set for interactive learning of visually grounded word meanings. In *Proceedings of the Sixth Workshop on Vision and Language*, pages 1–10. Association for Computational Linguistics.

Yu, Y., Lemon, O., and Eshghi, A. (2015b). Interactive learning through dialogue for multimodal language grounding. In *SemDial 2015, Proceedings of the 19th Workshop on the Semantics and Pragmatics of Dialogue, Gothenburg, Sweden, August 24-26 2015*, pages 214–215.

Yu, Y., Lemon, O., and Eshghi, A. (2016d). Comparing dialogue strategies for learning grounded language from human tutors. In *SemDial 2016, Proceedings of the 20th Workshop on the Semantics and Pragmatics of Dialogue, New Brunswick, NJ, USA,, July 16-18, 2016*, pages 44–54.

Zhang, H., Xiao, X., and Hasegawa, O. (2014). A Load-Balancing Self-Organizing Incremental Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6):1096–1105.

Zhang, M. and Zhou, Z. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM.